PARTS OF THE ORIGINAL DOCUMENT

ARE OF POOR REPRODUCTIVE QUALITY

CERTAINES PARTIES DU DOCUMENT ORIGINAL

SONT DE PIÈTRE QUALITÉ DE REPRODUCTION

UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, STATION A, MONTRÉAL, QUÉBEC, H3C 3J7

GENERATION DES PARCOURS POUR LE TRANSPORT SCOLAIRE

par
Pierre A. Forgues

Cette recherche a été rendue possible grâce à une subvention du Centre de recherche et de développement de Transport Canada et à une subvention du programme F.C.A.C. du Ministère de l'Education du Québec.

Université de Montréal

Centre de recherche sur les transports - Publication # 123
Département d'informatique et de recherche opérationnelle
Publication # 315

Juin 1978

Dépôt légal - 1er trimestre 1979

Résumé

Ce document traite d'une adaptation des heuristiques standard de transport de marchandises à la génération des parcours unidirectionnels d'autobus pour le transport scolaire en milieu urbain où les étudiants sont rassemblés à des arrêts. Les heuristiques de Clarke & Wright [C6], de Gillet & Miller [G2], de Newton & Thomas [N1,N2], de Yellow [Y1] et d'Insertion sont décrits et les détails propres aux parcours d'autobus sont mentionnés. Une adaptation au transport scolaire de l'algorithme de post-optimisation par échanges inter-parcours de type Lin-2 [L2] est aussi décrite. Des programmes ont été développés pour chaque heuristique et algorithme, ils ont servis à résoudre quinze problèmes dont les résultats ont permis de déduire une méthode d'approche qui tient compte des facteurs cout-performance. Ce document contient le code de ces programmes et la description des cartes de contrôle et fichier pour l'ordinateur CYBER-173 de l'Université de Montréal. Une approche heuristique permettant de générer des parcours sur les segments du milieu est proposée en vue d'une recherche ultérieure.

Remerciements

L'auteur remercie son directeur de stage, monsieur Jacques A. Ferland, Ph.D., professeur agrégé, pour son assistance et ses conseils. Il remercie également messieurs Jean-Marc Rousseau, Ph.D., Jacques Desrosiers, M.Sc., Guy Lapalme, M.Sc., et Gilles Dulac, pour leur support technique et méthodologique, de même que monsieur B.L. Golden pour nous avoir fourni son programme de la méthode de Clarke & Wright duquel nous avons extrait quelques modules. L'auteur remercie particulièrement la Commission Scolaire Régionale St-François en monsieur Lucien Bolduc, régisseur au transport scolaire, pour nous avoir procuré les données nécessaires à la génération des parcours et pour ses critiques constructives des résultats.

TABLE DES MATIERES

Chapitre		Page
1.	Introduction au problème de génération des parcours	1
2.	Méthodologie	2
3.	L'Implantation de la génération des parcours	13
4.	Méthode de Clarke & Wright et extensions à la méthode de Yellow	23
5.	Méthode de Gillet & Miller et extensions à la méthode de Newton & Thomas	34
6.	Méthode d'insertion	42
7.	Algorithmes d'amélioration de parcours	49
8.	Description des fichiers	67
9.	Les étapes informatiques de la génération des parcours	77
10.	Tableaux et analyse des résultats	115
11.	Le problème général sur les arrêts et segments.....	136
12.	Conclusion	147

Annexe		
I	Liste des procédures informatiques.....	153
II	Liste des instructions source des programmes.....	155
III	Liste sommaire des résultats obtenus.....	278

TABLE DES PRINCIPALES ILLUSTRATIONS

Illustrations		Page
2.1	Décomposition en points de transferts	6
2.2-A	Routes générées par un heuristique de "dispatching"	11
2.2-B	Parcours d'autobus dans le milieu québécois	11
3.1	Cheminement choix d'arrêts-génération de parcours	18
4.1	Etapes dans l'heuristique de Clarke & Wright	25
7.1	Analyse des échanges avec un OPT-2	51
7.2	Représentation interne des inversions	51
7.3	Les quatre types d'échanges d'un OPT-3	54
7.4	Construction du cycle à partir de plusieurs parcours	57
7.5	Les parcours sous forme d'arbre	59
7.6	Analyse des échanges entre parcours	62
7.7	Echanges analysés entre P_i et P_j	64
10.1	Description des problèmes traités	116
10.2	Description des heuristiques et paramètres utilisés	117
10.3	Sommaire des solutions heuristiques	119
10.4	Tableau des solutions choisies	123
10.5	Tableau des deux "meilleurs" heuristiques	124
10.6	Complexité des heuristiques	125
10.7	Approche au problème	126

10.8 Heuristique à post-optimiser et choix du paramètre λ 128

10.9 Sommaire des solutions heuristiques post-optimisées 131

10.10 Tableau des solutions post-optimisées choisies 134

10.11 Comparaison avec / sans post-optimisation 135

11.1 Transformation d'un problème sur arrêts et segments en un problème sur les segments 138

1. Introduction au problème de la génération des parcours dans le transport scolaire

Le problème de la génération des parcours dans le transport scolaire se place dans le cadre d'une étude à long terme qui couvre le choix des arrêts, l'assignation des étudiants à ces derniers, la génération des parcours et l'agencement des parcours dans le temps.

On retrouve sous le thème de génération de parcours, deux problèmes distincts: le problème sur les arrêts en milieu urbain et le problème plus général sur les arrêts et segments en milieu rural. Notre étude traite en profondeur le premier problème et propose une méthode d'attaque au second.

Nous retrouvons dans le texte une discussion des particularités du problème en milieu québécois, la description détaillée des heuristiques adaptés à notre implantation et les phases informatiques de la réalisation. Un échantillon de quinze problèmes réels, fourni par une commission scolaire, ont servi à analyser les heuristiques et à déduire une méthode d'approche dans une implantation à grande échelle.

Dans les trois premiers chapitres, le lecteur est introduit à la problématique du transport scolaire. Les chapitres quatre à six décrivent les heuristiques utilisés dans notre implantation, le chapitre sept élabore sur les procédures de post-optimisation. Les chapitres huit et neuf traitent des aspects informatiques de la génération. Le chapitre dix analyse les parcours obtenus à partir de problèmes réels. Le chapitre onze propose deux heuristiques pour le problème plus général sur arrêts et segments et le dernier chapitre conclue par une discussion des objectifs que nous avons atteints.

2. Méthodologie

Cette section relate la méthodologie employée pour traiter le problème de génération des parcours dans le transport scolaire. Cette description est abordée en deux temps: l'identification du problème, la recherche et l'élaboration de méthodes de solution.

2.1 Identification du problème

2.1.1 Parcours vs circuits

De prime abord il serait bon de différencier entre les termes génération de parcours et création de circuits. Un circuit est un agencement dans le temps de plusieurs parcours alors qu'un parcours est un chemin permettant à un véhicule d'une certaine capacité de transporter les élèves vers l'école ou vers un point de transfert et vice versa, et ce sans situation temporelle.

2.1.2 Définition du problème

Notre problème se définit comme suit: étant donné un nombre d'étudiants assignés à des arrêts, et supposant que ceux-ci doivent être rassemblés en un lieu que nous nommerons l'école, générer des parcours respectant les contraintes de capacité, de distance et de nombre d'arrêts, tel que tous les étudiants soient transportés vers l'école et que la distance totale parcourue par les véhicules soit minimum.

2.1.3 Rassemblement initial aux arrêts

Dans le milieu urbain, les étudiants peuvent circuler en bordure des voies publiques, nous pouvons donc les rassembler en des points que nous nommerons arrêts.

Ce rassemblement initial permet, lorsque bien localisé, de réduire le nombre d'autobus requis au transport. De plus, il peut aussi permettre de réduire la distance de parcours. Ce rassemblement a donc un impact direct sur le coût du transport scolaire.

Il existe une forte relation entre la distance de marche à pied des étudiants et le nombre d'arrêts. Plus la distance est grande, plus le nombre d'arrêts peut être petit. Nous n'explorons toutefois pas ce problème, puisqu'il fait déjà l'état d'une autre étude. Nous supposons donc que les arrêts sont pré-déterminés et que le point de rassemblement est connu.

2.1.4 Non-unicité du point de rassemblement

Le point de rassemblement que nous nommons "école" peut ne pas être unique dans un problème. Ceci survient lorsqu'un régisseur regroupe le transport de plusieurs écoles rapprochées, afin de permettre des parcours communs. Les étudiants des dites écoles sont transportés dans les mêmes véhicules et les parcours se terminent par un tour partiel ou complet des écoles.

Nous pouvons générer des parcours avec cette contrainte supplémentaire en posant l'hypothèse suivante: les parcours doivent se terminer par une visite de chacune des écoles. Assumant ceci, nous pouvons diviser le problème en deux: visiter les écoles et générer les parcours. Pour chacune des écoles, nous devons connaître le quasi-tour minimum de visite des autres écoles, ainsi que le coût de chacun. Nous pouvons alors évaluer pour chaque arrêt, la meilleure école à se rendre pour minimiser le coût de visite supplémentaire.

Connaissant pour chaque arrêt la distance vers chaque école, et pour chaque école, la distance minimum du tour des autres écoles et la séquence. Nous évaluons pour chaque arrêt la distance arrêt-destinations comme suit:

$$\text{dist (arrêt, destinations)} = \text{Min} \begin{bmatrix} \text{dist (arrêt, école 1)} + \text{dist. (tour partant de l'école 1)} \\ . \\ . \\ . \\ \text{dist (arrêt, école n)} + \text{dist. (tour partant de l'école n)} \end{bmatrix}$$

L'avantage de cette technique est de rendre transparent la présence de plusieurs destinations et de pouvoir utiliser les heuristiques standards de génération de parcours. Les modules de sortie des parcours auront à donner la séquence de visite des écoles selon le dernier arrêt du parcours. La détermination des tours des écoles devenant indépendante des parcours, elle peut être faite dans une étape préalable.

2.1.5 Le milieu rural

Bien que nous ne traitons pas le milieu rural, il est toutefois bon de savoir qu'il présente une organisation différente. Les étudiants ne sont plus rassemblés à des arrêts, ils sont embarqués devant leurs domiciles. La faible densité de population, aussi bien que la non-carrosabilité des voies publiques par les piétons exigent cette contrainte. Dès qu'un segment possède un étudiant (ou plus), nous devons le parcourir.

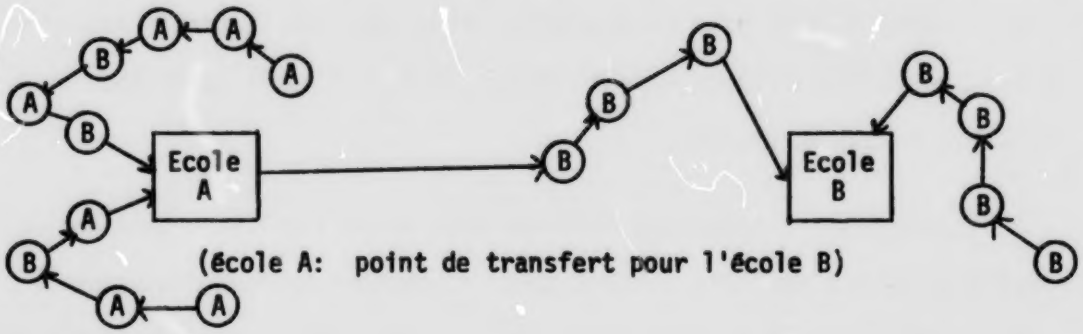
Voyons maintenant le problème reliant les deux approches: soit parcourir des segments et traverser un village où plusieurs arrêts existent. Ce problème général sur les arrêts et segments peut être modifié pour devenir un problème sur les segments uniquement. Ce problème n'est pas

explicitement traité dans cette étude. Nous exposons toutefois dans une autre section une méthodologie d'attaque basée sur notre expérience du problème en milieu urbain.

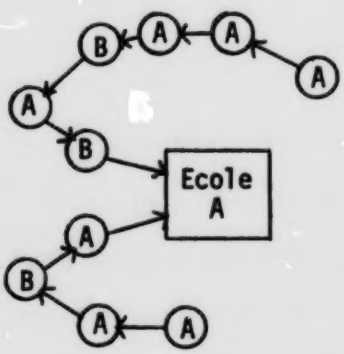
Il est aussi à noter que certains étudiants du milieu urbain doivent être embarqués à domicile. Parmi ceux-ci mentionnons des élèves de la pré-maternelle et de la maternelle, et les handicapés. Ces cas particuliers doivent être traités à l'aide des algorithmes spéciaux au milieu rural, étant donné la similitude du problème.

2.1.6 Les points de transferts

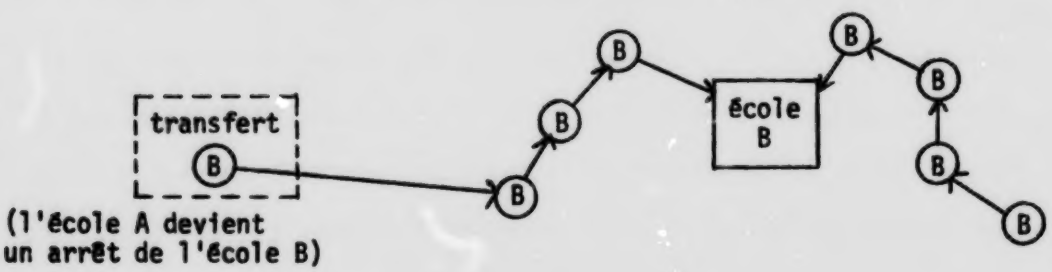
Dans notre approche, nous traitons individuellement chaque point de rassemblement. Un point de rassemblement peut être une ou plusieurs écoles ou un point de transfert. Le point de transfert est utilisé pour exécuter un rassemblement régional des étudiants. Une fois rendus à ce point, les étudiants sont redistribués vers leurs écoles respectives. Le point de transfert est principalement utilisé pour le milieu rural. Ces étudiants sont traités en deux étapes. Dans la première étape ils sont rassemblés au point de transfert. Ce point est considéré comme une école fictive. Dans un second temps, le point de transfert devient un arrêt pour chacune des écoles où nous avons des étudiants à distribuer. La figure 2.1 explique cette décomposition lorsqu'une école sert de point de transfert.



Circuits des écoles A et B



Parcours école A



Parcours école B

Fig. 2.1 Décomposition en points de transfert

La partie du bas de la figure donne la décomposition en point de transfert du problème original du haut de la page.

2.2 Recherche et élaboration des méthodes de solution

2.2.1 Le réseau des arrêts

Notre réseau de transport G(N,A) est constitué d'un noeud de destination, celui-ci représente une ou plusieurs écoles, lorsque nous traitons plusieurs écoles, ce noeud est celui de la première. S'ajoutent à celui-ci, les noeuds des (N-1) arrêts. Pour chaque arrêt, nous connaissons la distance requise pour se rendre aux destinations. Nous connaissons en plus pour chacun des arrêts, la distance les reliant à un nombre d'arrêts avoisinants, la distance entre deux arrêts non-reliées est infinie. Pour chaque arrêt, nous connaissons le nombre d'étudiants assignés à transporter.

Pour solutionner le problème, nous devons connaître les divers types d'autobus disponibles et pour chaque type, leur nombre et leur capacité. Il nous faut aussi les contraintes de distance et de nombre d'arrêts maximums d'un parcours, celles-ci sont données globalement pour le problème. Nous avons maintenant les données nécessaires pour identifier le problème.

2.2.2 Solutions globalement ou localement optimales

La définition du réseau des arrêts nous indique que notre problème en est un sur les arrêts. Lorsqu'un seul véhicule est utilisé, alors le problème est similaire à celui du commis-voyageur. Selon Desrosiers [D1], "une récente recherche montre que le problème du commis-voyageur est complexe et qu'un algorithme borné polynomialement n'existe pas, c'est-à-dire que le nombre d'opérations requis pour le résoudre n'est pas borné,

par exemple, par n^2, n^3,... En d'autres mots, pour obtenir la meilleure solution de façon certaine au problème du commis-voyageur, il faut énumérer toutes les solutions possibles et choisir celle qui a le coût (distance parcourue) minimum".

Lorsque le problème à un seul véhicule est généralisé à n véhicules, chacun devant obéir à des contraintes de distance, de nombre d'arrêts et de capacité, le seul objectif de minimiser la distance totale des véhicules est insuffisant. Parfois une solution nécessitant un nombre inférieur de véhicules peut être préférée à une autre dont la distance totale de parcours est inférieure mais nécessitant un plus grand nombre de véhicules. Pour un régisseur, la meilleure solution en est une qui pondère la distance totale et le nombre de parcours. Or, ce qui est bon pour un régisseur ne l'est pas pour l'autre.

La difficulté de quantifier l'optimalité d'une solution à plusieurs véhicules et l'absence d'algorithmes exacts nous font rejeter la recherche de solutions globalement optimales. Il existe cependant des méthodes approximatives ou heuristiques qui sélectionnent les possibilités et qui procurent de très bonnes solutions localement optimales.

2.2.3 Les approches au problème

Tous les heuristiques de génération de parcours peuvent être regroupés sous l'une des grandes approches: "grouper en premier, circuler en second" et "circuler en premier, grouper en second".

L'approche "grouper en premier, circuler en second", découpe le territoire en sous-territoires qu'un seul véhicule peut desservir en

respectant les contraintes de distance, de capacité et de nombre d'arrêts. Les méthodes de Clarke & Wright [C6] , de Yellow [Y1] et d'Insertion peuvent être classées sous ce chapitre.

L'approche "circuler en premier, grouper en second", résout le problème comme s'il n'y avait qu'un seul véhicule à capacité infinie, obtenant ainsi un gigantesque parcours que l'on brise ensuite en parcours réalisables. La méthode de Newton & Thomas [N1] est un exemple de cette approche.

La méthode de Gillet & Miller [G2] assume la présence d'un grand tour sans explicitement le calculer. Elle se trouve donc à mi-chemin entre les deux approches.

Nous discutons les détails de ces heuristiques dans les sections suivantes. Notons que ces heuristiques génèrent des solutions initiales, qui peuvent être localement optimales ou non. Pour le vérifier, nous utilisons des algorithmes qui tentent de diminuer la distance totale des parcours, par des échanges de deux ou trois segments. Ces méthodes dûes à S. Lin [L2] et mieux connues sous les noms Lin-2 et Lin-3, génèrent des solutions qui sont respectivement optimales de niveau 2 (OPT2) et de niveau 3 (OPT3).

2.2.4 Adaptation des heuristiques au transport scolaire

Les heuristiques considérés pour résoudre le problème de génération des parcours pour le transport scolaire sont des adaptations des heuristiques de "dispatching" pour le problème de livraison par camions. Chacun

de ces parcours se débute et se termine à l'entrepôt. Ceci se rapproche beaucoup du problème de m commis-voyageur. La figure 2.2.-A décrit cette situation.

Par contre, dans le transport scolaire québécois, le point de départ du parcours n'est pas nécessairement le même que la destination. La distance qui sépare le garage et le premier arrêt du premier parcours a un coût nul. En effet, les coûts de location des autobus sont définis comme suit: les salaires des chauffeurs pour des périodes de quatre ou huit heures et un coût variable proportionnel à la distance des circuits. Or, étant donné que l'agencement final des parcours n'est pas connu, on peut négliger la distance entre l'endroit d'où origine l'autobus et le premier arrêt du parcours. La situation est illustrée à la figure 2.2-B.

L'application des techniques de transport de marchandises au transport scolaire peut s'envisager de deux façons: la première et la plus simple consiste à appliquer directement ces heuristiques en posant l'hypothèse que les parcours débutent et se terminent à l'école. Or, cette hypothèse ne semble pas trop réjouissante à tout régisseur d'expérience puisque l'expérience a déjà été tentée dans un organisme québécois, avec le produit VSPX [Il] (une implantation de l'heuristique Clarke & Wright par la firme IBM) sans obtenir de résultats satisfaisants.

La seconde méthode consiste à modifier ces heuristiques de façon à ce qu'ils n'aient pas d'origines préspécifiées pour les parcours, ceci revient à débuter chaque parcours à un arrêt artificiel qui a la propriété d'être à une distance nulle avec tous les arrêts réels. Les parcours

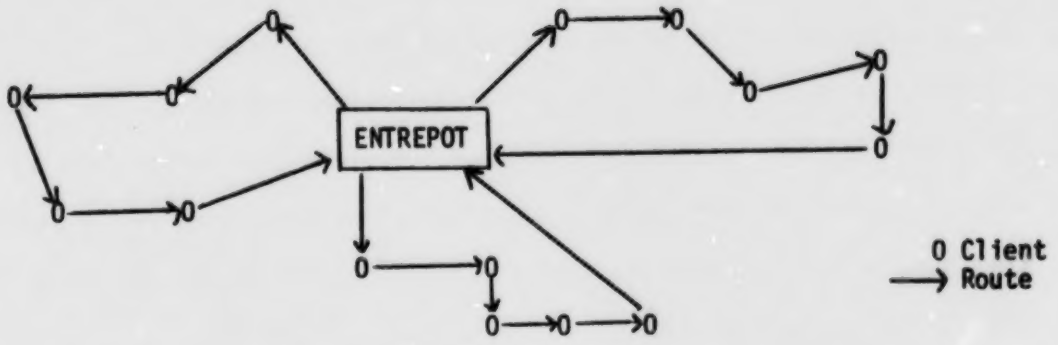


Fig. 2.2-A Routes générées par un heuristique de "dispatching"

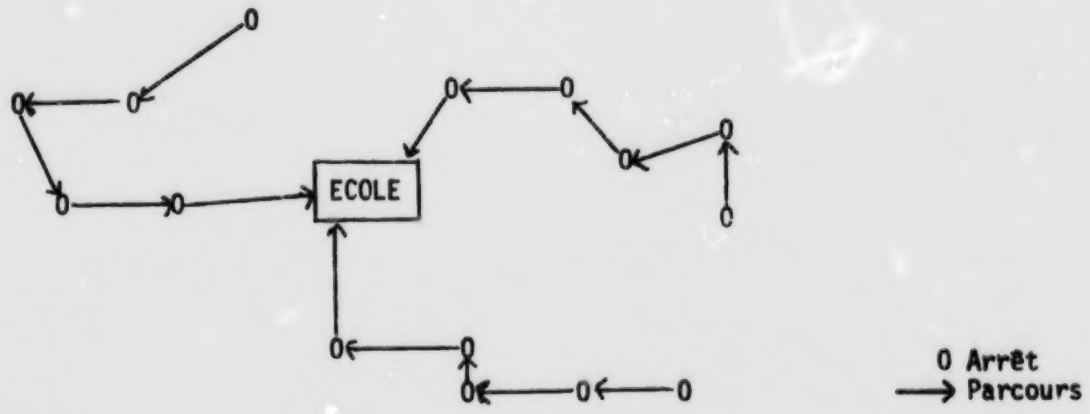


Fig. 2.2-B Parcours d'autobus dans le milieu québécois

générés à l'aide d'un tel arrêt auraient tendance à se diriger vers l'école en passant d'abord par les arrêts éloignés.

En termes opérationnels et sans augmenter le nombre d'arrêts, le même résultat peut être obtenu en fixant à zéro la distance de l'école à l'arrêt. La distance de l'arrêt à l'école demeure inchangée. Le réseau des arrêts devient orienté par rapport à l'école. Nous aurons donc à adapter les heuristiques à cette nouvelle contrainte. Nous verrons plus loin que les heuristiques les plus touchés sont ceux de post-optimisation par échanges inter-parcours.

3. L'implantation de la génération des parcours

3.1 Les modalités de son implantation

La discussion de l'implantation de la génération des parcours ne peut se traiter unilatéralement sans introduire des concepts relatifs aux étapes préalables du choix des arrêts et de l'assignation des étudiants aux arrêts. Ainsi nous incorporons une discussion de la méthodologie de concertation entre ces étapes et la génération de parcours. Puis nous traitons de l'apport humain requis dans le choix de la solution et de son impact sur l'agencement des parcours dans le temps.

3.1.1 Sommaire du choix des arrêts

Le choix des arrêts peut se faire par une procédure informatique que le régisseur a le loisir d'intercepter et de modifier avant d'y assigner les étudiants. La procédure se décrit sommairement comme suit: Nous possédons une liste d'arrêts potentiels et une distance de marche maximum pour les étudiants assignés à un mini-arrêt. Un mini-arrêt est un des deux noeuds du segment sur lequel l'étudiant demeure. Pour chaque mini-arrêt avec des étudiants, nous déterminons les arrêts auxquels nous pouvons assigner les étudiants compte-tenu de la contrainte de marche. Pour chaque arrêt potentiel nous cumulons le nombre d'étudiants qui peut s'y regrouper. Une procédure itérative est alors entreprise. L'arrêt avec le plus grand potentiel de regroupement est choisi, et tous les étudiants des mini-arrêts regroupés sont déduits des autres arrêts où ils étaient cumulés, affectant ainsi le regroupement aux autres arrêts potentiels. La procédure est répétée avec quelques finesses pour les cas d'égalité, et

ce, jusqu'à ce que chaque étudiant soit assigné à un arrêt. La procédure du choix d'arrêts peut être interrompue par le régisseur afin d'y apporter des modifications pour tenir compte de certaines particularités difficiles à quantifier. D'une part la procédure du choix des arrêts a tendance à simuler le travail du régisseur en localisant les arrêts dans des zones relativement denses. D'autre part, la procédure cherche à minimiser le nombre d'arrêts, en générant des arrêts où un grand nombre d'étudiants sont rassemblés, ce qui n'est pas nécessairement l'objectif de tous les régisseurs. La seule variable qu'on peut modifier pour augmenter le nombre d'arrêts est la contrainte de marche. Plus la distance de marche est petite, plus le nombre d'arrêts est grand. Nous verrons plus loin comment le régisseur peut parer à ces restrictions.

3.1.2 Sommaire de l'assignation des étudiants aux arrêts

Le choix des arrêts étant réalisé, nous devons maintenant y assigner les étudiants. Le problème se résout en deux étapes. Dans la première, les étudiants sont assignés à l'arrêt le plus prêt de leur mini-arrêt respectif. Si on introduit des contraintes de capacité aux arrêts, alors dans une seconde étape, on doit résoudre un problème de transport. Malheureusement le problème peut devenir non-réalisable si le nombre d'arrêts est trop petit. En effet, il peut arriver qu'un trop grand nombre d'étudiants doivent être assignés à un seul arrêt qui représente la seule possibilité pour ces derniers, relativement à la contrainte de distance de marche maximum.

Nous verrons plus loin que le problème de capacité peut être attaqué d'un autre point de vue.

3.1.3 Les arrêts dans la génération des parcours

Les parcours que nous créons sont directement dépendants des deux étapes du choix et de l'assignation des étudiants aux arrêts décrites préalablement. Notons que celles-ci font totalement abstraction du concept de parcours. Bien qu'il soit impossible (si ce n'est manuellement) de produire simultanément un choix d'arrêts et de parcours, nous nous devons de donner au régisseur des outils lui permettant d'intervenir dans le choix des arrêts et dans la génération des parcours, et peut-être même de concerter ces étapes.

Les heuristiques de génération de parcours, de par leurs concepts, créent des solutions réalisables à partir des arrêts et du nombre d'étudiants s'y trouvant spécifiés. Un arrêt est intégré à un parcours si le nombre d'étudiants n'excède pas la capacité du véhicule. Si cette contrainte ne peut être respectée, l'arrêt n'est pas choisi et sera intégré à un autre parcours. Ce sont des procédures "tout ou rien" dans le sens qu'on ne peut intégrer partiellement un arrêt à un parcours en transportant une portion de ses étudiants s'y trouvant. Ces heuristiques peuvent donc générer des parcours particulièrement mauvais lorsque le nombre d'étudiants à chaque arrêt est grand; les parcours sont alors allongés et s'entrecoupent de façon inacceptable.

Un outil a été développé dans le but de contrer à ce problème et de permettre au régisseur d'intervenir dans une relocalisation des arrêts après analyse des parcours. Il s'agit du découpage préalable des arrêts. Le régisseur peut spécifier le nombre maximum d'étudiants qu'il désire

à un arrêt. Lorsque le nombre d'étudiants excède cette quantité, l'arrêt est alors découpé en plusieurs arrêts, et les étudiants y sont uniformément distribués. Les arrêts ainsi créés sont dits artificiels dans le sens qu'ils duplicatent l'arrêt original.

Prenons l'exemple d'un arrêt de 25 étudiants, une contrainte de 10 étudiants par arrêt est demandée. Le traitement créera donc 3 arrêts. Les deux premiers de 8 étudiants et le dernier de 9. Deux de ceux-ci seront artificiels. Des arcs sont créés pour relier les arrêts artificiels à l'arrêt original, pour relier les arrêts artificiels entre eux, pour relier les arrêts artificiels à tous les arrêts que l'original pouvait rejoindre et même pour relier des arrêts artificiels d'un arrêt aux arrêts artificiels d'un autre arrêt. Cette prolifération du nombre d'arcs supplémentaires nous indique que l'utilisation de cette technique de duplication devrait être limitée. Une distance de 1 mètre est assumée entre l'arrêt réel et ses arrêts artificiels.

Le découpage des arrêts permet d'uniformiser le nombre d'étudiants aux arrêts et de réduire dans certains cas les distances de parcours en offrant plusieurs choix aux heuristiques. L'arrêt découpé peut donc faire partie d'autant de parcours différents que de fois qu'il a été duplicaté et peut aussi bien être regroupé sous le même parcours.

Lorsqu'un arrêt original a été duplicaté et qu'il appartient à plusieurs parcours, le régisseur a tout loisir de déplacer l'arrêt artificiel à un noeud rapproché sur le parcours du véhicule sans augmenter la distance du trajet. Cette même opération peut aussi être faite sur un

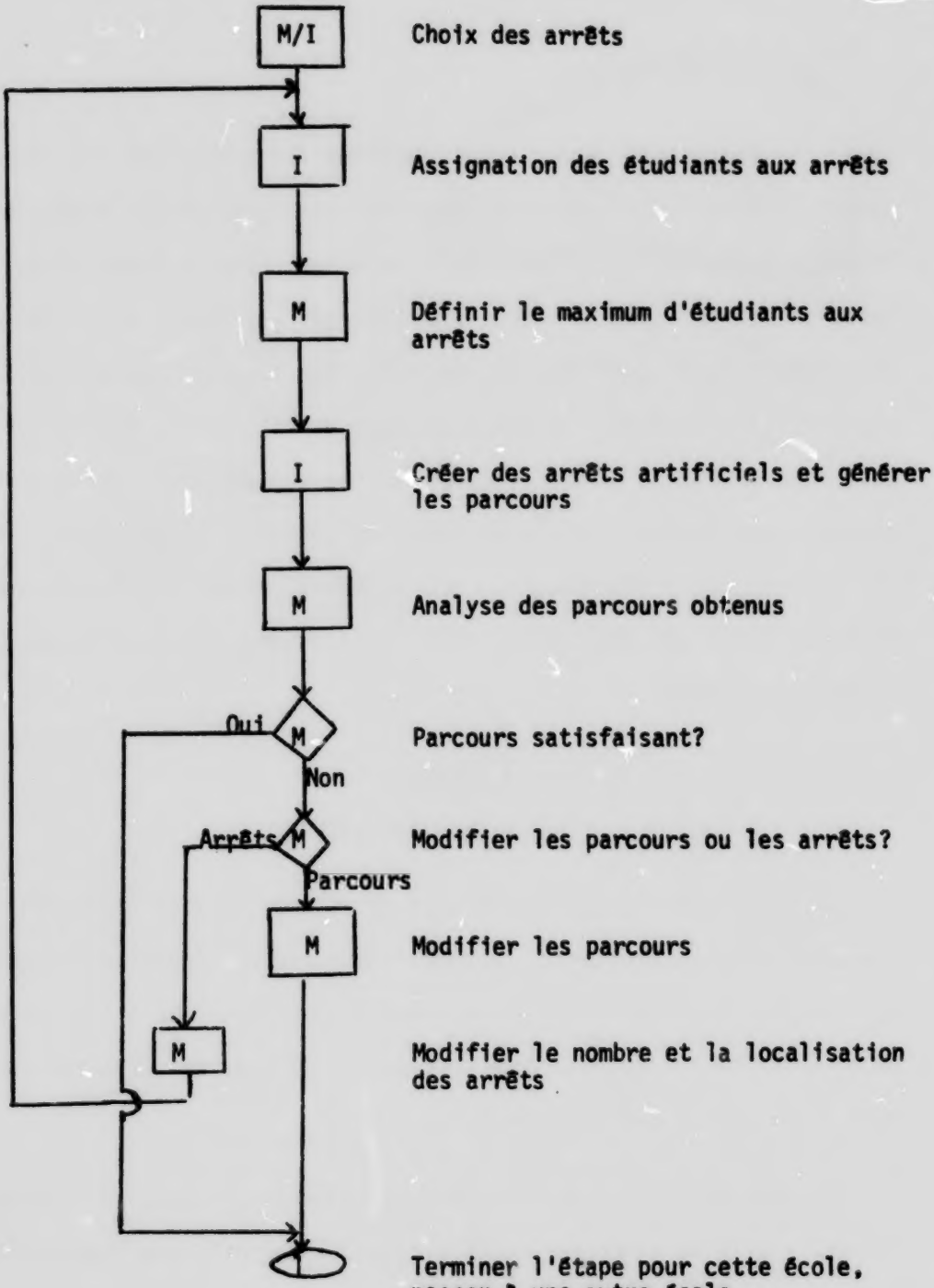
arrêt non duplicaté. Après avoir distribué ses arrêts sur les parcours générés, le régisseur peut commander une réassignation des étudiants aux nouveaux arrêts en leur fixant cette fois une capacité d'étudiants, la solution de l'assignation étant donc généralement réalisable. Les parcours sont regénérés et on répète la même opération jusqu'à l'obtention de parcours satisfaisants. Le processus est décrit dans la figure 3.1. Il doit donc exister un nombre d'étudiants maximum par arrêt permettant une génération de parcours se rapprochant des objectifs du régisseur. Un tel nombre n'est malheureusement pas universel et varie selon le problème et les objectifs du régisseur. Nous laissons donc à l'utilisateur le choix de ce nombre.

3.2 Choix des parcours à agencer

3.2.1 Evaluation du meilleur parcours

Les heuristiques considérés dans notre implantation produisent soit une ou plusieurs solutions. Le nombre de parcours généré aussi bien que la distance totale des parcours varient d'un heuristique à l'autre et d'une solution à l'autre. Nous devons donc établir des critères permettant d'évaluer la meilleure solution.

Premièrement, nous avons établi que pour deux solutions comportant le même nombre de parcours, celle produisant une distance totale minimum est préférable. Puis nous avons conclu que deux solutions avec un nombre de parcours différent ne pouvaient être comparées sur le seul critère de distance totale. Nous conservons donc la meilleure solution en termes de distance totale pour chaque nombre de parcours.



M: opération manuelle I: opération informatique

Fig. 3.1. Cheminement choix d'arrêts - génération de parcours

Par exemple, si nous obtenons des solutions de 9, 10 et 11 parcours, nous conservons pour chacun des nombres de parcours, la meilleure solution en terme de distance totale. Nous n'avons plus qu'à choisir entre la meilleure des 9, 10 et 11 parcours. Dans notre implantation, la solution choisie est celle minimisant le facteur parcours-distance. On multiplie le nombre de parcours par la distance totale des parcours pour obtenir ce facteur. Cette solution sera écrite sur un fichier en vue de l'agencement des parcours. Ce critère ne faisant peut-être pas l'unanimité de tous les régisseurs, nous imprimons la meilleure solution de chaque nombre de parcours, lui permettant donc d'intervenir dans ce choix.

3.2.2 Liens avec l'agencement des parcours

La dernière phase du système concerne l'agencement des parcours générés, modifiés ou non par le régisseur pour tenir compte de facteurs imponderables. Bien que nous ne pouvons traiter en quelques lignes de la fonction la plus ardue du système, mentionnons ces objectifs.

L'agencement des parcours a pour but d'assigner les parcours à des véhicules dans le temps dans le but d'en minimiser le nombre. Un ou plusieurs parcours peuvent être réalisés par un même véhicule. Il s'agit de déterminer l'heure de début et fin de chaque parcours et de créer des circuits journaliers de véhicules. Pour réaliser ceci, l'horaire des écoles doit varier. Une école doit débuter tôt et l'autre plus tard. Plus l'horaire des écoles peut être élastique et déplaçable, plus le nombre de véhicules à utiliser aura tendance à être petit. L'agencement se fait à l'intérieur d'un cadre horaire "au plus tôt et au plus tard" et

avec un ensemble de contraintes sur les temps d'arrivée-départ, de millage improductif et autres. Pour une étude plus approfondie du problème, on réfère le lecteur au texte de J. Desrosiers [D1] .

3.2.3 Relation parcours-véhicules

La création de circuits agence les parcours de façon à minimiser le nombre de véhicules requis. Le nombre de parcours à agencer ne détermine pas le nombre de véhicules requis. Certains parcours sont plus difficiles à agencer. Une solution de la génération de parcours avec N parcours peut produire un moins bon agencement qu'une solution avec N+1 parcours. Des parcours de courtes distances, même si le nombre est plus grand, peuvent peut-être mieux s'agencer. Les possibilités d'agencement sont proportionnelles au nombre de parcours et à la distance requise pour relier un parcours à un autre. Il peut toutefois exister des cas où un nombre restreint de parcours s'agence mieux.

Il nous est impossible au moment de la génération de parcours de prévoir le couplage. Nous devons donc nous contenter d'optimiser localement pour décider des solutions qui seront utilisées dans la confection de circuits.

3.3 Remarques sur les contraintes

Les contraintes sont perçues par les heuristiques comme des frontières à la génération de parcours. Un arrêt est ajouté au parcours seulement si les contraintes de nombre d'arrêts, de distance et de capacité sont respectées; si une des contraintes est violée, l'arrêt est laissé pour

un autre parcours. Voyons comment chacune des contraintes est évaluée dans les heuristiques.

La contrainte du nombre d'arrêts d'un parcours est violée si le nombre d'arrêts réels et artificiels est plus grand que la limite fixée. En fait, plus le nombre d'étudiants par arrêt est petit, plus la contrainte du nombre d'arrêts doit être grande. Notons enfin que la génération des parcours en milieu urbain ne devrait pas être contrainte par le nombre d'arrêts.

La contrainte de distance est violée si la distance du parcours excède la limite fixée. Lorsque nous traitons le problème sans origine à l'école, la distance est celle du trajet partant du premier arrêt et se rendant à l'école. Tandis que dans le problème avec origine et destination à l'école la distance couvre le trajet partant de l'école, visitant les arrêts dans la séquence déterminée et retournant à l'école. Dans le second cas, la contrainte de distance doit être plus grande que dans le premier cas. En milieu urbain, la distance ne devrait pas être contraignante. En milieu rural, la distance maximum d'un parcours est normalement atteint avant que l'autobus soit rempli à capacité.

La contrainte de capacité est respectée si un autobus de capacité suffisante peut être assigné au parcours. Lorsque plus d'un type d'autobus peut être utilisé, celui qui minimise le nombre de sièges libres est assigné pour laisser les autobus de plus grande capacité disponibles. Lorsqu'un arrêt doit être ajouté à un parcours, l'autobus qui y était assigné est rendu disponible avant d'analyser la contrainte de capacité.

Si aucun autobus ne peut être assigné au nouveau parcours, l'ancien autobus est réassigné à l'ancien parcours et l'arrêt n'est pas inclu.

En milieu urbain, la capacité des autobus est généralement atteinte avant que les contraintes de nombre d'arrêts et de distance interviennent. De ce fait, les autobus de grande capacité sont d'abord utilisés. Le régisseur doit donc être conscient de ces particularités avant d'entreprendre la définition des contraintes.

4. Méthode de Clarke & Wright et extensions à la méthode de Yellow

Cette section présente l'heuristique de Clarke &Wright [C6] grandement utilisé dans les travaux de génération de routes pour le transport de marchandises. Cet heuristique est la base du produit VSPX [I1] de la firme IBM. Nous présentons également une extension de cet algorithme dûe à Yellow [Y1].

4.1 Clarke & Wright

4.1.1 Introduction à la méthode de Clarke & Wright

L'heuristique de base de Clarke & Wright consiste à créer initialement autant de parcours qu'il y a d'arrêts. Il s'agit par la suite de réduire le nombre de parcours en reliant deux parcours existant pour n'en former qu'un seul. Le choix des deux parcours à fusionner est fonction d'une épargne calculée par la formule suivante:

$$S_{ij} = d_{1i} + d_{1j} - d_{ij} \qquad (4.1)$$

Nous réduisons la distance totale parcourue du facteur S_{ij} lorsque nous relions les arrêts i et j, situés aux extrémités de deux parcours différents. La distance de l'école à l'arrêt i(d_{1i}) et à l'arrêt j (d_{1j}) est sauvée, mais la distance entre les arrêts i et j (d_{ij}) est encourue. Cette épargne (S_{ij}) est calculée pour tous les arcs reliant des arrêts, une fois au début de la procédure, elles sont triées en ordre décroissant puis examinées de la plus grande à la plus petite. L'arc i-j est utilisé si il sert à relier deux parcours différents de façon que leur union respecte les contraintes de nombre d'arrêts, de capacité et de distance de

parcours. La figure 4.1 décrit les étapes de ce processus.

4.1.2 Faiblesses et améliorations

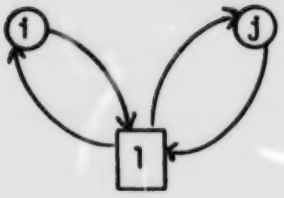
L'heuristique tel que décrit n'offre qu'une seule solution et a le désavantage de ne pas défaire les parcours qu'il a créés. Les arcs sont examinés dans l'ordre du meilleur au moins bon potentiel de réduction de distance totale, pour être intégrés dans les parcours. Une fois qu'un arc est choisi et intégré, il y demeure définitivement. La création des parcours par réduction, c'est-à-dire par l'union itérative de deux parcours existants est une technique sous-optimale. En effet, il est possible et même fréquent d'obtenir trois petits parcours qui auraient pu en former deux si les arcs avaient été examinés dans un ordre différent.

L'action prise pour contrer ces problèmes est de générer plusieurs solutions à partir du même problème en variant un paramètre (λ) dans la formule du calcul des épargnes. La nouvelle formule (4.2) ne présente plus les diminutions de la distance totale, mais plutôt une mesure de désirabilité pour chaque arc d'être incorporé dans les parcours.

$$S_{ij} = d_{1i} + d_{ij} - \lambda d_{ij} \qquad (4.2)$$

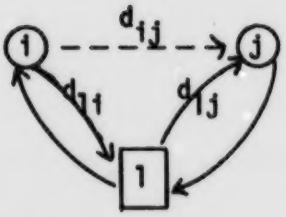
(λ: paramètre qui lorsque varié altère la forme des parcours)

L'auteur qui le premier a utilisé ce paramètre est Yellow [Y1], celui-ci n'a pas uniquement modifié le calcul des S_{ij} mais il a en plus modifié la construction des parcours. Nous verrons les détails de son travail plus loin dans cette section. Remarquons dans la formule 4.2 que lorsque λ vaut 1, nous retrouvons la technique originale. Dans le



école

chaque arrêt forme un parcours



d_{ij}

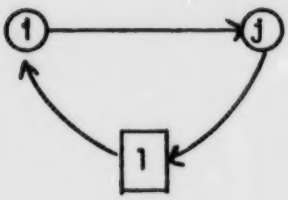
d_{1i}

d_{1j}

école

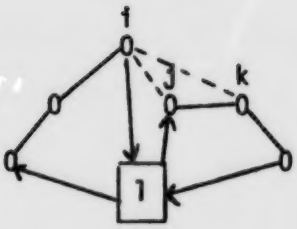
pour chaque arc i-j, on calcule

$S_{ij} = d_{1i} + d_{1j} - d_{ij}$



école

on relie les parcours dont les épargnes sont les meilleures



école

dans cette figure, seul l'arc i-j peut être utilisé, l'arc i-k devient non réalisable puisqu'il ne conserve pas la forme des parcours, k n'est pas un sommet extrême du parcours auquel il appartient

Fig. 4.1 Etapes dans l'heuristique de Clarke & Wright

cas qui nous concerne, les parcours à générer ne doivent pas tenir compte de la distance de l'école au premier arrêt. Une façon d'intégrer cette particularité de notre problème consiste à imputer à chacun des arcs arrêt-école et école-arrêt la moitié de la distance séparant l'arrêt de l'école. Notons que cette modification est justifiable sous l'hypothèse que les distances arrêts-écoles sont presque uniformes. L'approche se reflète dans la formule d'épargne (4.3). Si on multiplie par deux l'expression (4.3), on retrouve alors un ordonnancement des épargnes qui est le même que celui qu'on aurait obtenu avec la formule (4.2) en posant $\lambda = 2$. Ceci est assumé dans (4.4). Puisque la distance arrêt-école n'est pas uniforme dans les problèmes réels, nous devrons solutionner pour plusieurs valeurs de λ, autour de 2.

$$S_{ij} = \tfrac{1}{2} d_{1i} + \tfrac{1}{2} d_{1j} - d_{ij} \qquad (4.3)$$

$$S_{ij} = d_{1i} + d_{1j} - 2d_{ij} \qquad (4.4)$$

En somme notre problème même avec sa particularité peut être traité par l'algorithme de Clarke & Wright avec une valeur de λ appropriée.

$$S_{ij} = d_{1i} + d_{1j} - \lambda d_{ij} \qquad (4.5)$$

4.1.3 Formule avec pénalité

Dans le but d'améliorer les solutions obtenues par la méthode de Clarke & Wright, nous avons tenté plusieurs modifications de la formule (4.2) pour le calcul des S_{ij}. On en est arrivé à une formule incluant un facteur de pénalité encouru lorsque l'arc (i,j) ne se dirige pas vers

l'école. Le raisonnement suivant nous permet d'en déduire la validité lorsque les parcours ont avantage à se diriger vers l'école.

Compte tenu du fait que les distances inter-arrêt calculées sont minimum et que la loi de triangle est respectée, nous pouvons déduire (4.6).

$$\begin{aligned} d_{1i} + d_{ij} &\geq d_{1j} \\ d_{1j} + d_{ij} &\geq d_{1i} \end{aligned} \qquad (4.6)$$

Examinons maintenant la pénalité (4.7) et montrons que celle-ci est une mesure de directionnalité.

$$P = |d_{1i}-d_{1j}| - d_{ij} \qquad (4.7)$$

Si $d_{1i} > d_{1j}$ alors $d_{1i}-d_{1j} > 0$. Or selon (4.6)

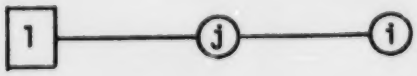
$d_{1i}-d_{1j} \leq d_{ij}$, et par conséquent

donc $$d_{1i}-d_{1j}-d_{ij} \leq 0 \qquad (4.8).$$

Analysons maintenant les deux cas de (4.8).

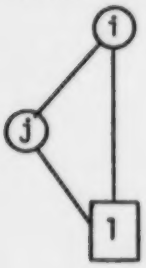
<u>Cas 1</u>: $$d_{1i}-d_{1j}-d_{ij} = 0 \qquad (4.9)$$

On en déduit facilement que pour aller à i, partant de l'école 1, nous devons passer par j.

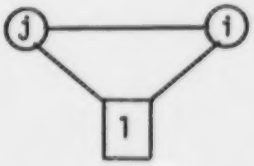


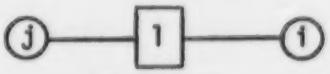
<u>Cas 2</u>: $$d_{1i}-d_{1j}-d_{ij} < 0 \qquad (4.10)$$

Pour aller de l'école à i, nous ne passons pas par j. Nous remarquons toutefois que plus la valeur de 4.10 se rapproche de zéro, plus l'arc (i,j) se dirige vers l'école. Les trois figures suivantes illustrent ce fait.



p petit





p très petit

Le même raisonnement peut être appliqué au cas où $d_{lj} > d_{li}$. La formule 4.7 constitue donc une pénalité valable. La formule 4.11 est celle utilisée dans notre implantation. Nous avons laissé le choix à l'usager de demander l'une ou l'autre des formules 4.5 ou 4.11.

$$S_{ij} = d_{li} + d_{lj} - \lambda d_{ij} + [\,|d_{li}-d_{lj}| - d_{ij}\,] \qquad (4.11)$$

Nous verrons dans la section suivante que notre heuristique a besoin d'épargnes S_{ij} positifs. Or, les formules (4.5) et (4.11) peuvent générer des S_{ij} négatifs ou nuls, pour contrer ceci nous additionnerons aux formules (4.5) et (4.11) une constante positive M pour nous assurer des S_{ij} positifs. Les formules (4.5) et (4.11) deviennent donc (4.5)' et (4.11)' respectivement.

$$S_{ij} = d_{li} + d_{lj} - \lambda d_{ij} + M \qquad (4.5)'$$

$$S_{ij} = d_{li} + d_{lj} - \lambda d_{ij} + [\,|d_{li}-d_{lj}| - d_{ij}\,] + M \qquad (4.11)'$$

4.1.4 Etapes logiques de l'heuristique de Clarke & Wright

Etapes	Description
[A]	Pour chaque arc (i,j) où $i \neq 1$ et $j \neq 1$, calculer les S_{ij} selon la formule (4.5)' ou (4.11)' avec λ fixé.
[B]	Trier les arcs (i,j) en ordre décroissant de S_{ij} (Dans notre implantation ceci a été réalisé à l'aide d'un "heap sort" en s'inspirant de l'implantation qu'a fait B. Golden dans [G4]).
[C]	Choisir l'arc (i,j) tel que S_{ij} est le plus grand. Si $S_{ij} = 0$, aller à l'étape [H].
[D]	- Si i et j appartiennent au même parcours, aller à l'étape [G] - Si i et j ne sont pas tous deux des arrêts extrêmes de leur parcours respectifs, aller à l'étape [G] - Si le nombre d'étudiants résultant de l'union des deux parcours excède la capacité des véhicules disponibles, aller à l'étape [G] - Si le nombre d'arrêts résultant de l'union des deux parcours excède la contrainte de nombre d'arrêts, aller à l'étape [G].
[E]	On calcule la distance résultant de l'union des deux parcours. Si nous traitons le problème sans origine pré-fixée pour les parcours, nous imputons de la distance calculée, la distance arrêt-école de l'arrêt extrême le plus éloigné de l'école. - Si la distance résultant excède la contrainte de distance maximum pour un parcours, aller à l'étape [G].
[F]	- Unir les deux parcours avec l'arc (i,j) - Mettre S_{ij} à zéro

Etapes	Description
	- Si i n'est plus un arrêt extrême du nouveau parcours, rendre tous les S_{ik} et S_{ki} à zéro.
	- Si j n'est plus un arrêt extrême du nouveau parcours, rendre tous les S_{jk} et S_{kj} à zéro
	- Réarranger les (i,j) tel que les S_{ij} sont en ordre décroissant (ceci se réalise rapidement lorsque les S_{ij} sont sous un "HEAP")
	- Aller à l'étape [C].
[G]	- Mettre le S_{ij} à zéro
	- Réarranger les (i,j) tel que les S_{ij} sont en ordre décroissant
	- Aller à l'étape [C].
[H]	Nous avons terminé de construire les parcours
	- Réoptimiser la séquence de visite des arrêts à l'intérieur de chaque parcours à l'aide d'un OPT2.
[I]	Terminer

4.2 Yellow

4.2.1. Les modifications de l'heuristique de Clarke & Wright par Yellow

Outre la modification de la formule d'épargne (4.2) Yellow [Y1] a réalisé une transformation fondamentale à l'algorithme de Clarke & Wright, dans le but de palier au trop grand nombre de parcours souvent générés. De par sa conception, l'heuristique de Clarke & Wright construit plusieurs parcours simultanément. Initialement, un autobus est assigné à chaque

arrêt. Au fil des itérations, le nombre de parcours diminue et le nombre d'arrêts par parcours croît. Deux parcours peuvent s'unir si les contraintes de nombre d'arrêts, de distance parcourue et de charge sont respectées. La méthode fait que certains parcours ne peuvent s'unir sans violer une des contraintes, ceux-ci demeurent donc à la fin, accroissant le nombre de parcours résultant.

La méthode de Yellow a pour but premier de minimiser le nombre de parcours et pour but second de minimiser la distance totale. Il construit donc un seul parcours à la fois en utilisant les S_{ij} calculées à partir des formules (4.5)' ou (4.11)' comme mesure de désirabilité d'un arc d'entrer dans le parcours. Il choisit initialement le meilleur S_{ij} et débute avec un parcours formé de l'arc (i,j). Puis itérativement, il prolonge le parcours par ses extrémités par choix des meilleurs S_{ij} possible. Lorsqu'il n'existe plus d'arcs (i,j) à incorporer au parcours sans violer les contraintes, le parcours est considéré final. Le processus est repris pour un autre parcours.

4.2.2. Etapes logiques de l'heuristique de Yellow

Etapes	Description
[A]	Pour chaque arc (i,j) où $i \neq 1$ et $j \neq 1$, calculer les S_{ij} selon la formule (4.5)' ou (4.11)'.
[B]	Trier les arcs (i,j) en ordre décroissant de S_{ij} (Dans notre implantation, ceci a été réalisé à l'aide d'un heap-sort).
[C]	Choisir l'arc (i,j) avec le plus grand S_{ij} - Si le parcours formé de l'arc (i,j) est réalisable, aller à [D]

Etapes	Description
	- Sinon . mettre ce S_{ij} à zéro
	. réordonner les arcs (i,j) en ordre décroissant de S_{ij}
	. aller à [C]
[D]	- Unir les arrêts i et j pour former le parcours
	- Mettre ce S_{ij} à zéro
	- Réordonner les (i,j) en ordre décroissant de S_{ij}
[E]	Choisir le prochain arc (i,j) dont le S_{ij} est le plus élevé
	- Si ce S_{ij} vaut 0, aller à [I]
[F]	- Si cet arc (i,j) peut être raccordé à une des extrémités du parcours en construction, aller à [G] (ceci est vérifié si i ou j est lui-même une extrémité du parcours en construction)
	- Sinon . mettre $S_{ij} = - S_{ij}$
	. réordonner les arcs (i,j) en ordre décroissant de S_{ij}
	. aller à [E]
[G]	- Mettre S_{ij} à zéro
	- Réordonner les arcs (i,j) en ordre décroissant de S_{ij}
	- Si i et j appartiennent au même parcours, aller à [E]
	- Si le nouveau parcours formé par l'extension du parcours actuel avec l'arc (i,j) est réalisable aller à [H]
	- Sinon aller à [E]
[H]	- Incorporer l'arc (i,j) au parcours actuel
	- Si i n'est plus un arrêt extrême du nouveau parcours, rendre tous les S_{ik} et S_{ki} à zéro

Etapes	Description
	- Si j n'est plus un arrêt extrême du nouveau parcours, rendre tous les S_{jk} et S_{kj} à zéro
	- Réarranger les (i,j) en ordre décroissant de S_{ij}
	- Aller à [E]
[I]	- On a terminé de construire ce parcours, l'enregistrer de façon permanente
	- Pour tous les arcs (i,j) mettre $S_{ij} = - S_{ij}$
	- Ordonner tous les arcs (i,j) en ordre décroissant de S_{ij}
	- Si au moins un S_{ij} est positif ($S_{ij} > 0$), aller à [C]
	- Sinon, tous les S_{ij} sont nuls, tous les parcours sont construits, aller à [J]
[J]	- Réoptimiser la séquence de visite des arrêts à l'intérieur de chaque parcours à l'aide d'un OPT-2
[K]	Terminer

5. Méthode de Gillet & Miller et extensions à la méthode de Newton & Thomas

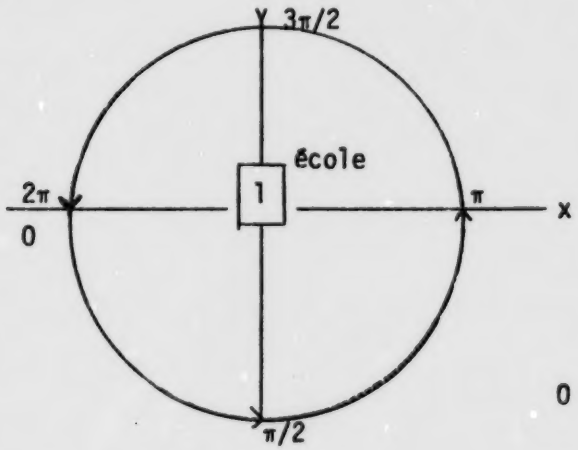
5.1 Gillet & Miller

5.1.1 Introduction à la méthode de Gillet & Miller

L'heuristique de Gillet & Miller [G2] mieux connu sous le nom de "SWEEP" utilise la position des arrêts dans le plan, relative à l'école pour partitionner son ensemble d'arrêts en sous-ensembles pour lesquels les parcours sont générés.

Les coordonnées X-Y de l'école et de chaque arrêt sont connues. Nous calculons donc les coordonnées polaires de chaque arrêt CP_i, en utilisant l'école comme origine dans le plan. L'angle et le rayon sont combinés selon la formule (5.1) pour obtenir la position dans l'espace de chaque arrêt.

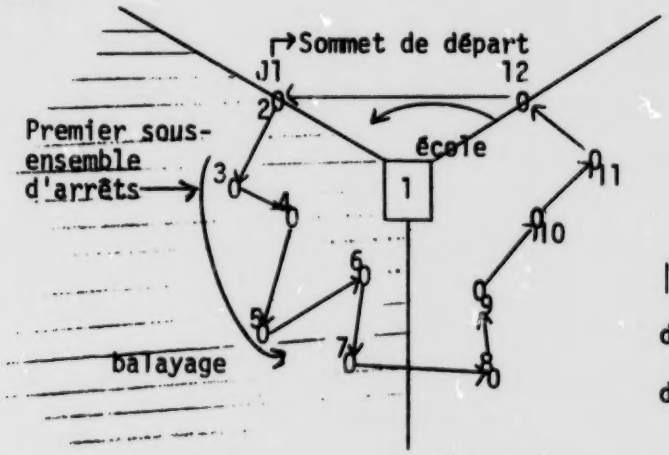
$$CP_i = (ANGLE_i \times 1\ 000\ 000) + RAYON_i \qquad (5.1)$$



$0 \leq angle < 2\pi$

Les sommets i sont ensuite triés en ordre croissant de CP_i, le dernier sommet est relié au premier pour ainsi former un grand tour passant

par tous les arrêts. Un sommet de départ J1 est pris comme étant un des sommets de la paire des sommets consécutifs dans la suite pour laquelle la différence de leur CP_i est maximum. Ceci est illustré dans la figure suivante



$|CP_2-CP_{12}|$ est maximum donc J1 = 2 est le sommet de départ choisi

En choisissant ensuite les arrêts en débutant par J1, en ordre de CP, nous déterminons un sous-ensemble d'arrêts avec lesquels nous générons un parcours initial. Ce parcours est ensuite analysé afin de voir si l'on pourrait échanger un arrêt de présent parcours contre un et deux arrêts du prochain sous-ensemble en réduisant notre distance de parcours, tout en ne pénalisant pas le prochain parcours. La procédure détaillée est présentée dans une section ultérieure.

Le plus grand handicap de la méthode de Gillet & Miller est de ne pas considérer les particularités physiques du réseau des arrêts dans la construction initiale du grand tour par les coordonnées polaires. Dans la méthode originale, le choix du sous-ensemble d'arrêts pourrait produire un parcours de distance infinie lorsqu'un obstacle comme une rivière rend le sous-réseau non connexe. Nous avons partiellement surmonté ce problème en s'assurant de la connexité du sous-ensemble. Ainsi à l'entrée de

chaque nouvel arrêt nous résolvons un OPT-2 avec les arrêts choisis, et si la distance de parcours est infinie, nous passons à un autre arrêt sans choisir celui-ci. Cette particularité dû à la connaissance partielle de la matrice des distances ne va pas sans altérer de rendement de l'heuristique. Les remarques ci-haut, appuyées par des résultats auto-explicatifs indique que cet heuristique ne semble pas tellement approprié pour générer des parcours dans le transport scolaire.

5.1.2 Etapes de l'heuristique de Gillet & Miller

Etapes	Description
[A]	- Calculer la coordonnée polaire CP_i selon la formule (5.1) - Calculer AVR = la moyenne des rayons de chaque arrêt - Ordonner les arrêts dans le vecteur S_i en ordre croissant de CP_i - Choisir le premier sommet J1 du découpage dans le vecteur S tel que décrit précédemment - Aller à [B]
[B]	Définir J l'arrêt de début de la liste des arrêts hors-parcours dans le grand tour des coordonnées polaires. Si il n'existe plus d'arrêts hors-parcours, aller à [U]. Former un premier parcours en reliant l'école et l'arrêt J. Retirer J de la liste des arrêts hors-parcours.
[C]	Rechercher parmi les 10 arrêts du début de la liste, le premier arrêt J qui si intégré au parcours en construction conserve sa réalisibilité en termes de distance, de nombre d'arrêts et de capacité.

Etapes	Description
	(La distance de parcours est évaluée en intégrant temporairement l'arrêt au parcours et en minimisant la distance de parcours à l'aide de l'algorithme OPT-2 qui est décrit à la section 7.11). Si aucun des 10 arrêts ne peut être intégré au parcours, aller à l'étape [D] .
	Si un arrêt J peut être intégré au parcours, Intégrer l'arrêt J et le retirer de la liste des arrêts hors-parcours. Aller à l'étape [C] .
	Si on a épuisé la liste des arrêts hors-parcours, aller à l'étape [T].
[D]	Calculer la distance D1 pour le parcours créé, à l'aide de l'algorithme OPT-2
[E]	Déterminer l'arrêt JJX hors-parcours le plus rapproché du dernier arrêt du sous-ensemble
[F]	Déterminer l'arrêt JII hors-parcours le plus rapproché de JJX
[G]	Déterminer KII appartenant au sous-ensemble tel que KII maximise rayon (KII) + angle (KII) * AVR (KII est un arrêt loin de l'école et près du prochain parcours)
[H]	Déterminer la distance D2 du parcours obtenu en ajoutant JJX en retirant KII. Si l'inclusion de JJX viole la contrainte de capacité, aller à [J]
[I]	Si la distance D2 respecte la contrainte de distance, alors aller à [K]
[J]	Enregistrer le parcours actuel formé par le sous-ensemble aller à [B]

Etapes	Description
[K]	Déterminer la distance D3 pour parcourir les six arrêts suivant le sous-ensemble, partant et terminant à l'école
[L]	Déterminer la distance D4 pour parcourir les mêmes six arrêts qu'à l'étape [K] sauf que JJX est remplacé par KII dans ces arrêts. Si JJX n'appartient pas à l'un de ces six arrêts, aller à [J]
[M]	Si D1 + D3 < D2 + D4 aller à [O]. Sinon aller à [N]
[N]	Retirer KII du sous-ensemble et le remplacer par JJX. Mettre KII au sommet de la liste des arrêts hors parcours, y retirer JJX mettre J = KII Aller à l'étape [B] pour augmenter le sous-ensemble
[O]	Evaluer la distance D5 pour le parcours dont KII a été remplacé par JJX et JII. Si l'inclusion de ces deux sommets viole la contrainte d'arrêts ou de capacité, aller à [J]. Si JJX et JII ne sont pas dans les six arrêts suivant le sous-ensemble, aller à [J]
[P]	Si la distance D5 viole la contrainte de distance, aller à [J]
[Q]	Déterminer D6 la distance du parcours qui inclue KII et les six arrêts suivant le sous-ensemble, mais qui exclue JJX et JII
[R]	Si D1 + D3 < D5 + D6 aller à [J]
[S]	Retirer KII du sous-ensemble et le remplacer par JJX et JII. Mettre KII au sommet de la liste des sommets hors-parcours, y retirer JJX et JII. Mettre J = KII.

Aller à l'étape [B] pour augmenter le sous-ensemble

[T] Calculer la distance D1 pour le parcours créé à l'aide de l'algorithme OPT-2

Enregistrer le parcours

[U] Terminer

5.2 Newton & Thomas

5.2.1 Introduction à la méthode de Newton & Thomas

La méthode de Newton & Thomas [N1] vise à surmonter des handicaps de la méthode de Gillet & Miller [G2] en tenant explicitement compte de la structure du réseau. La méthode consiste à générer dans un premier temps un grand tour passant par tous les arrêts avec une distance aussi petite que possible. Le grand tour est ensuite découpé en parcours réalisables.

Dans notre implantation, le grand tour est généré à l'aide de l'heuristique de Clarke & Wright en supposant qu'un véhicule de capacité infinie est disponible et qu'il n'y a pas de restrictions de distance ni de nombre d'arrêts sur le parcours. Le grand parcours débute et termine à l'école. Le problème est solutionné quatre fois à l'aide de la formule (4.5) avec le paramètre λ respectivement égal à 0.8, 0.9, 1.0 et 1.1. La meilleure des quatre solutions est refaite et utilisée comme grand tour. L'arrêt de début de découpage est le premier arrêt visité partant de l'école. Il semble évident que meilleur est le grand tour initial, meilleure sera la solution finale.

La procédure de découpage utilisée dans notre implantation est plus complexe que celle proposée par Newton & Thomas dans le sens qu'on y a incorporé des étapes de "look ahead" tel que proposé par BODIN [B3].

La particularité de la méthode est de former le moins de parcours possible, laissant souvent peu d'étudiants au dernier parcours. La qualité des parcours générés est proportionnelle à l'effort mis pour solutionner le problème du commis-voyageur de création de grand tour. Notons toutefois que le meilleur grand tour peut ne pas produire de meilleurs parcours compte tenu du découpage. Nous avons tenté de pondérer ces deux facteurs en cherchant un bon grand tour sans être le meilleur.

5.2.2 Etapes logiques de l'heuristique de Newton & Thomas

Etapes	Description
[A]	Générer S un grand tour partant de l'école, passant par chaque arrêt et revenant à l'école, à l'aide de l'heuristique Clarke & Wright [C6] auquel on permet des véhicules de capacité infinie sans contraintes de distance ou d'arrêts. Conserver le ou les parcours dans le vecteur S en les reliant l'un à l'autre. Le vecteur S constitue par le fait même un ordonnancement des arrêts.
[B]	Définir le premier arrêt J où débute ou se poursuit le découpage du grand tour. Si il n'existe plus d'arrêts hors-parcours, aller à l'étape [G]

Etapes	Description
[C]	Former un premier parcours en reliant l'école et J, retirer J de la liste des arrêts hors-parcours
[D]	Rechercher parmi les 10 arrêts du début de la liste, le premier arrêt J qui si intégré au parcours en construction conserve sa réalisibilité en termes de distance, de nombre d'arrêts et de capacité. (La distance de parcours est évaluée en intégrant temporairement l'arrêt au parcours et en solutionnant un OPT-2). Si aucun des 10 arrêts ne peut être intégré au parcours, aller à l'étape [F]. Si on a épuisé la liste des arrêts hors-parcours, aller à l'étape [F].
[E]	Intégrer l'arrêt J au parcours et le retirer de la liste des arrêts hors-parcours. Aller à l'étape [D]
[F]	Le parcours est maintenant complet. Optimiser le parcours des arrêts avec une procédure OPT-2, enregistrer le parcours. Aller à l'étape [B] si il existe d'autres arrêts hors-parcours
[G]	Terminer

6. Méthode d'insertion

Les heuristiques d'insertion pour résoudre les problèmes de commis-voyageur sont nombreux. Mentionnons par exemple l'insertion la plus rapprochée, la plus éloignée, arbitraire et enfin celle du "convex hull" toutes traitées dans le texte de Golden [G3]. Les résultats obtenus avec ces heuristiques sur des problèmes à un véhicule se comparent favorablement avec ceux préalablement mentionnés. Ceux-ci sont même meilleurs dans certains problèmes. Leur complexité est toutefois mauvaise car elle est de $O(n^2 \log_2 n)$ où n est le nombre d'arrêts.

Dans cette section, nous traitons de notre adaptation d'un de ces heuristiques d'insertion au problème à plusieurs véhicules. Celle-ci semble bien simuler l'approche manuelle du régisseur tout en construisant le moins de parcours possible.

6.1 Introduction à l'heuristique d'insertion

L'heuristique d'insertion pour générer des parcours à plusieurs véhicules est une adaptation du problème à un véhicule. Les principes régissant le problème à un seul véhicule sont décrits dans les étapes ci-après.

Etapes	Description
[A]	Créer un cycle initial en reliant l'école à l'arrêt le plus éloigné
[B]	Pour chaque arrêt K hors-cycle, trouver l'arc (i,j) appartenant au cycle, tel l'expression A_{kij} dans (6.1) soit minimum. Conserver ce triplet (k,i,j)

Etapes	Description
[C]	Choisir le triplet (k^*,i^*,j^*) parmi les triplets (k,i,j) tel que $A_{k^*i^*j^*}$ soit minimum
[D]	Insérer k^* entre i^* et j^* dans le cycle, retourner à l'étape [B]. Si l'on a épuisé tous les arrêts hors-cycle, terminer

La formule d'évaluation A_{ikj} peut varier selon l'algorithme d'insertion, mais les étapes ci-haut demeurent inchangées. La formule (6.1) la plus souvent utilisée, est celle de l'insertion la moins coûteuse.

$$A_{ikj} = \min_{(i,j)} (d_{ik} + d_{kj} - d_{ij}) \qquad (6.1)$$

6.2 L'adaptation au problème à plusieurs véhicules

Dans l'adaptation de l'heuristique de base au problème à plusieurs véhicules nous voulions faire un seul balayage des arrêts hors-cycle en conservant des informations d'une itération à l'autre, et incorporer à l'heuristique résultant plus d'une formule d'évaluation d'insertion.

Pour déterminer le prochain arrêt hors-parcours à être intégré au parcours à l'aide d'un seul balayage, il suffit de choisir le triplet (k^*,i^*,j^*) comme étant le minimum sur tous les triplets (k,i,j) analysés. Les étapes [B] et [C] de l'algorithme de la section précédente peuvent donc être regroupés. D'autre part, dans le but d'utiliser l'information déjà disponible sur l'ordonnancement des triplets nous procédons comme suit. Dénotons par (k, i_k, j_k) le triplet ayant la plus petite valeur pour A_{ikj} pour chaque arrêt k hors-parcours à la dernière itération. Dénotons également k^* l'arrêt qu'on choisit d'intégrer au parcours entre

les arrêts i* et j* à la dernière itération.

Pour chaque arrêt k hors-parcours nous analysons les deux cas suivants:

Cas 1. Si le couple (i_k,j_k) est le couple (i^*,j^*), alors étant donné que k* se trouve maintenant entre i* et j* il s'ensuit que le segment reliant i* et j* n'existe plus et par conséquent il faut réévaluer les quantités A_{ikj} pour tous les segments (i,j) du parcours actuel pour déterminer le triplet (k,i_k,j_k).

Cas 2. Si le couple (i_k,j_k) est différent du couple (i^*,j^*), alors il faut évaluer les quantités $A_{i^*kk^*}$ et $A_{k^*kj^*}$ et comparer ces valeurs avec $A_{i_jkj_k}$ pour déterminer le nouveau triplet (k,i_k,j_k).

Les cas qui sont valables lorsqu'on a un problème avec un seul véhicule doivent être analysés de nouveau lorsque plusieurs véhicules sont disponibles. L'introduction de l'arrêt k* dans le parcours peut prohiber l'insertion d'un autre arrêt k. C'est pourquoi avant d'insérer l'arrêt k il faut vérifier la réalisabilité du nouveau parcours en termes de distance, du nombre d'arrêts et de la capacité des véhicules.

Si la contrainte de nombre d'arrêts est violée, le parcours est complet. Si la contrainte de capacité est violée, k ne peut être inséré dans le parcours et ne doit plus être considéré pour ce parcours. Si la contrainte de distance est violée, k peut dans certains cas être inséré ailleurs dans le parcours, il doit alors être réexaminé au complet.

Dans notre implantation, nous avons considéré deux formules pour déterminer le triplet (k*,i*,j*), soit (6.2) et (6.3).

$$A_{i*k*j*} = \underset{(k,i,j)}{\text{Min}} \left[(d_{i,k} + d_{k,j})/d_{ij} \right] * d_{\ell,k} \qquad (6.2)$$

où $\ell = \begin{cases} i & \text{si } d_{1i} \geq d_{1j} \\ j & \text{si } d_{1i} < d_{1j} \end{cases}$

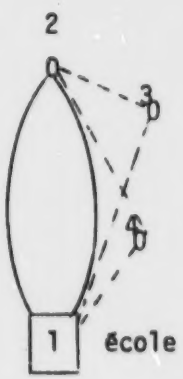
$$A_{i*k*j*} = -\lambda(\beta d_{1k} + d_{ij}) - d_{ik} - d_{kj} \qquad (6.3)$$

où λ est un paramètre permettant de varier les solutions.

et $\beta = \begin{cases} 1 & \text{lorsque les parcours se dirigent vers l'école} \\ 2 & \text{lorsque les parcours débutent et terminent à l'école} \end{cases}$

La formule (6.2) a pour effet de susciter des insertions minimisant la déformation du cycle et de choisir de préférence un sommet loin de l'école et près du cycle. Les parcours ainsi créés devraient être le plus directionnels possible. La figure 6.4 montre une situation simple.

fig. 6.4

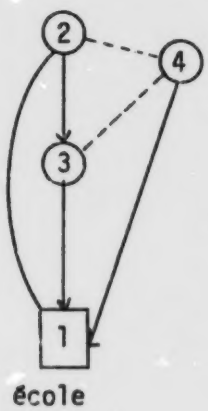


$A_{3.1.2}=(d_{13}+d_{23})/d_{12}*d_{23}$
$A_{4.1.2}=(d_{14}+d_{24})/d_{12}*d_{24}$
Supposons $(d_{13}+d_{23})=(d_{14}+d_{24})$
alors l'arrêt 3 est choisi pour entrer dans le cycle.

La formule (6.3) a une similitude avec la formule (4.1) dans la méthode de Clarke & Wright [C6]. Elle assume au départ que chaque arrêt est relié à l'école par un parcours. L'intégration de l'arrêt dans le parcours

a deux effets: augmenter la distance du parcours et sauver un parcours. L'heuristique s'approche donc de l'heuristique de Yellow [Y1], lequel construit un parcours à la fois en ajoutant des arrêts aux extrémités. Notre heuristique construit aussi un seul parcours à la fois, mais il permet d'intégrer un arrêt aussi bien aux extrémités qu'à l'intérieur du parcours. Il devrait donc dominer celui de Yellow. Son second avantage est d'être conçu pour traiter les parcours dirigés. Dans ce cas, le cycle n'est examiné que dans un seul sens. La figure (6.5) illustre ce fait.

Fig. 6.5



$S_{4,2,3}=-\lambda(d_{14}+d_{23})-d_{24}-d_{34}$

Dans le parcours résultant, on sauve les arcs (1,4) et (2,3) mais on ajoute les arcs (2,4) et (3,4)

6.3 Etapes logiques de l'heuristique d'insertion

Etapes	Description
[A]	Trouver parmi la liste des arrêts hors-parcours, l'arrêt L le plus éloigné de l'école
[B]	Former le cycle initial 1-L-1 si l'on traite le problème avec départ et arrivée à l'école ou le cycle L-1 si l'on traite le problème dans une seule direction.
	- Enlever L de la liste des arrêts hors-parcours.

Etapes Description

- Si nous avons épuisé la liste des arrêts hors-parcours, aller à l'étape [G]

[C] Annuler A_{i*k*j*}

Pour chacun des arrêts k hors-parcours, faire jusqu'à [F]

[D] Si k possédait une insertion réalisable à l'itération précédente des arrêts hors-parcours, alors

- vérifier si l'insertion de k demeure réalisable dans le nouveau parcours.
- Si la contrainte de nombre d'arrêts est violée aller à [H].
- Si la contrainte de capacité est violée ne plus considérer ce k pour le parcours en construction.
- Si la contrainte de distance est violée en introduisant k entre i_k et j_k
  - . annuler $A_{i_k k j_k}$
  - . Faire l'étape [E] pour tous les arcs (i,j) du parcours.
- Si l'insertion demeure réalisable
  - . faire l'étape [E] pour les arcs (i_k,j_k), (i^*,k^*) et (k^*,j^*)

Si k ne possédait pas d'insertion réalisable à la dernière itération, faire l'étape [E] pour les arcs (i^*,k^*) et (k^*,j^*).

[E] Parmi les arcs (i,j) spécifiés trouver le triplet (k,i_k,j_k) tel que la formule (6.2) ou (6.3) soit minimum et que l'insertion de k entre i et j respecte les contraintes de capacité, de distance et de nombre d'arrêts.

[F] Si $A_{i_k k j_k}$ est plus petit que A_{i*k*j*}, alors mettre $(k^*,i^*,j^*)=(k,i_k,j_k)$ et $A_{i*k*j*} = A_{i_k k j_k}$

Etapes	Description
	Itérer sur les arrêts k hors-parcours
[G]	Si l'insertion de k* n'est pas réalisable, aller à l'étape [H].
	Sinon . insérer k* entre i* et j* dans le cycle;
	. retirer k* de la liste des arrêts hors-parcours;
	. aller à l'étape [C].
[H]	Faire une procédure OPT-2 pour réduire la distance du parcours.
	Si on ne réduit pas la distance de parcours, aller à l'étape [I].
	Si on avait bloqué sur une contrainte de distance pour au moins un arrêt k et qu'on a pas déjà exécuté un OPT-2 à l'itération précédente, aller à l'étape [G].
	Sinon aller à l'étape [I]
[I]	Enregistrer le parcours actuel.
	Aller à l'étape [A] .
[J]	Terminer

7. Les algorithmes d'amélioration de parcours

Il ne suffit pas seulement de générer des parcours, mais reste encore à optimiser ceux-ci à un niveau désiré. Comme nous avons vu, la description des heuristiques inclue les termes OPT-2 et OPT-3 pour améliorer un parcours. Dans cette section, nous élaborons sur ces algorithmes d'optimisation en deux temps. Dans un premier temps nous traitons des améliorations à l'intérieur d'un seul parcours. Dans un second temps, nous généralisons ceux-ci à des échanges inter-parcours et nous y introduisons la description d'une méthode conçue spécialement pour le problème du transport scolaire. Les méthodes d'échanges inter-parcours que nous avons implanté se limitent au OPT-2, celui-ci étant déjà très couteux. Une application d'échange inter-parcours de type OPT-3 est décrite dans l'article de Russel [R1] mais celle-ci serait inabordable dans implantation à grande échelle.

7.1 Algorithmes d'amélioration d'un parcours

Le but de ces algorithmes est de réduire la distance d'un parcours en modifiant la séquence de visite des arrêts. Le problème est équivalent à résoudre de façon heuristique un problème de commis-voyageur. Les heuristiques les plus populaires proviennent de Croes [C5] pour le OPT-2, de Lin [L2] pour le OPT-3 et deLin & Kerninghan [L3] pour le OPT-k.
Notre implantation ne considère que les techniques OPT-2 et OPT-3. Le temps de solution de ces algorithmes est de complexité $O(n^k)$ où n est le nombre de sommets et k le degré d'optimalité requis OPT-k. En nous référant à l'article de Golden [G3] , une bonne solution initiale couplée

avec un OPT-2 ou OPT-3 nous amènent très près de l'optimum, dans un temps raisonnable. Voyons maintenant le fonctionnement de ces deux algorithmes et le niveau d'optimalité qui y est rattaché.

7.1.1 L'algorithme OPT-2

L'algorithme OPT-2 utilise une solution initiale, en l'occurence un parcours, et rend celui-ci optimal de niveau 2 en procédant à des échanges d'arcs, altérant ainsi la séquence de visite des arrêts. Tous les échanges de deux arcs contre deux autres sont analysés. Si un de ces échanges réduit la distance, le parcours est ainsi modifié et l'examen recommence. Les seuls échanges analysés sont ceux qui conservent l'unicité du cycle après échange.

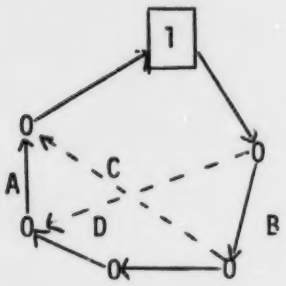
La méthode ne garantie nullement l'optimalité globale, mais elle garantie par contre qu'aucun échange de deux arcs puisse être réalisé pour réduire la distance. L'optimalité est donc locale de niveau 2. La figure (7.1) décrit un des échanges analysés.

La qualité de la solution est dépendante du cycle initial. L'espérance d'obtenir une très bonne solution avec un mauvais cycle initial est très mince. Le nombre de sommets influence beaucoup la qualité de la solution, puisque plus le nombre de sommets est grand, plus la probabilité de terminer sur un maximum local loin du global augmente[1]. Le court temps d'exécution de la technique et le fait que nos parcours comportent un nombre réduit d'arrêts rendent l'application de la technique d'autant plus appropriée dans notre étude. De plus, la technique ne nécessite

1 En fait, une solution à n sommets est optimale si elle est OPT-n, plus n est grand par rapport à 2, plus le OPT-2 a des chances de terminer loin de la solution OPT-n.

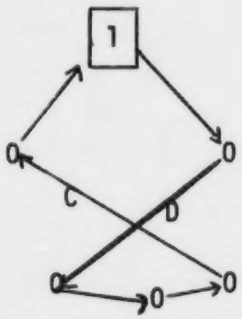
aucune modification pour supporter la directionnalité du réseau par rapport à l'école, celle-ci étant toujours le premier et dernier sommet du cycle.

Fig. 7.1 Analyse des échanges avec un OPT-2



AVANT ECHANGE

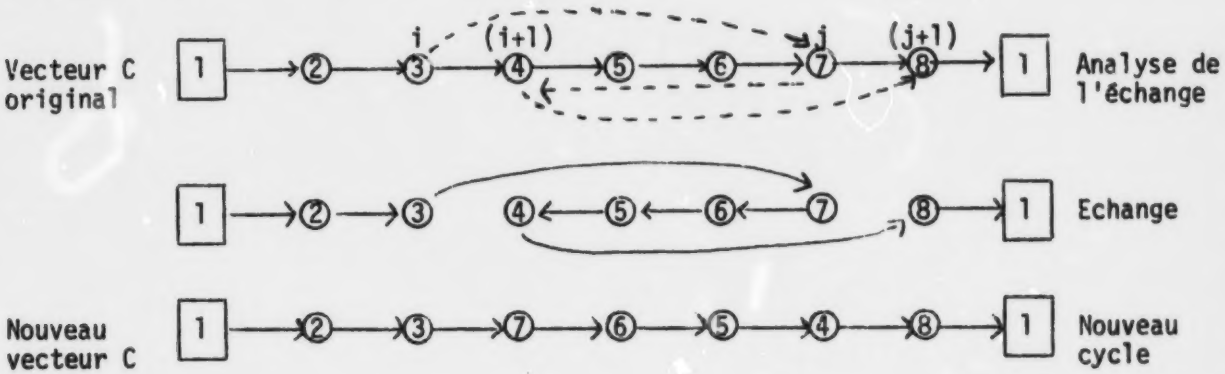
les arcs A et B sont analysés versus les arcs C et D



APRES ECHANGE

Après échange, la propriété de cycle est conservée

Fig. 7.2 Représentation interne des inversions



7.1.2 Etapes logiques de l'algorithme du OPT-2

Etapes	Description
[A]	Créer dans le vecteur C le cycle initial sous la forme suivante $C_1, C_2, C_3, \ldots, C_n, C_1$ (C_i contient le numéro du ième sommet dans le cycle)
[B]	Faire jusqu'à [G] pour $i = 1, \ldots, (n-2)$
[C]	Faire jusqu'à [F] pour $j = (i+1), \ldots, (n-1)$
[D]	Si $d_{C_i,C_j} + d_{C_{i+1},C_{j+1}} \geq d_{C_i,C_{i+1}} + d_{C_j,C_{j+1}}$ aller à [F]
[E]	On a trouvé un bon échange de deux arcs. - Inverser la portion (i+1), ..., (j) du vecteur C (la figure 7.2 montre cette inversion) - aller à [B]
[F]	Itérer sur j
[G]	Itérer sur i
[H]	Calculer la nouvelle distance du parcours
[I]	Terminer

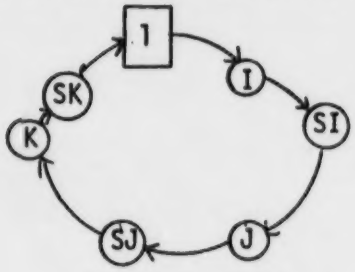
7.1.3 L'algorithme OPT-3

La procédure OPT-3 est une extension de la procédure OPT-2 dans le sens qu'elle examine les échanges non pas au niveau de deux arcs, mais bien de trois arcs. Cette procédure garantie une optimalité de niveau 3, et en utilisant la même solution initiale, cela doit engendrer une solution ayant une valeur plus petite ou égale à celle d'une solution obtenue avec une technique OPT-2.

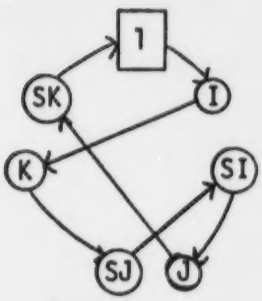
Comme dans le OPT-2, la qualité de la solution obtenue avec un OPT-3 est dépendante de la qualité du cycle initial. La complexité du OPT-3 est de n fois supérieure à celle du OPT-2, à savoir $O(n^3)$ où n est le nombre de sommets du cycle. Il existe deux versions de l'algorithme du OPT-3, la première, l'originale de Lin [L2] , demandant plus d'itérations que la version abrégée de Lin & Kerninghan [L3]. La méthode abrégée fait par contre plus d'analyse à chaque itération, ce qui rend ces deux méthodes de même complexité. Dans notre implantation, nous avons utilisé la dernière version puisque celle-ci était entièrement compatible avec la directionalité du réseau par rapport à l'école.

Dans l'algorithme du OPT-3, nous analysons tous les échanges de trois arcs contre trois autres. Pour ceci, quatre nouvelles représentations du cycle sont possibles. Chacune conserve la propriété cyclique après échange. Elles sont présentées dans la figure (7.3). La procédure est basée sur trois indices i, j, et k avec SI, SJ et SK leurs successeurs respectifs dans le cycle. Chacun des indices est varié de façon à analyser les quatre échanges possibles. Si le meilleur des échanges diminue la distance de parcours, il est réalisé et les itérations sont reprises au début avec le nouveau cycle.

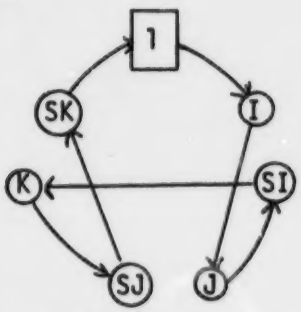
Fig. 7.3 Les quatre types d'échanges d'un OPT-3



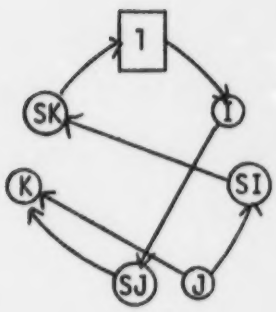
Cycle original



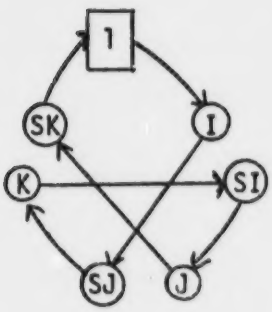
Type I



Type II



Type III



Type IV

7.1.4 Etapes logigues de l'algorithme OPT-3

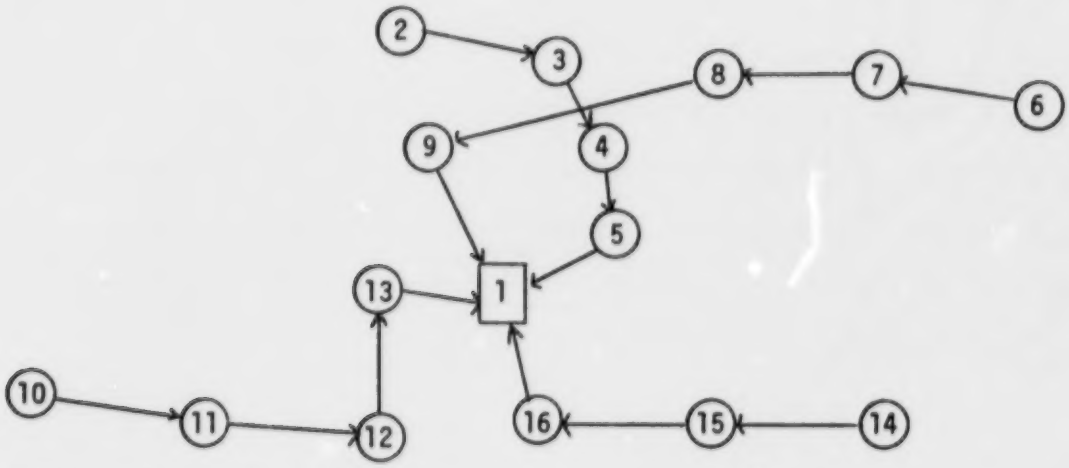
Etapes	Description
[A]	Créer dans le vecteur C, les n sommets du cycle sous la forme suivante: $C_1\ C_2\ C_3\ \ldots\ C_{n-1}\ C_n\ C_1$ (C_i contient le ième sommet du cycle)
[B]	Faire jusqu'à [J] pour i = 1, ... , n-2 identifier SI, le successeur de i dans le cycle
[C]	Faire jusqu'à [I] pour j = (i+1), ... , n-1 identifier SJ, le successeur de j dans le cycle
[D]	Faire jusqu'à [H] pour k = (j+1), ... , n identifier SK, le successeur de k dans le cycle
[E]	Identifier SAV_1, SAV_2, SAV_3 et SAV_4 le coût des échanges de type I, II, III et IV respectivement, selon la figure (7.3)
[F]	Choisir le meilleur des quatre échanges et mettre sa valeur dans SAV
[G]	Si le coût SAV diminue la distance par rapport au cycle original (i, SI), (j, SJ) et (k, SK), aller à [L]
[H]	Itérer sur k
[I]	Itérer sur j
[J]	Itérer sur i
[K]	Aller à [M]
[L]	Exécuter l'échange selon le type représenté par SAV aller à [B]
[M]	Calculer la nouvelle distance du parcours
[N]	Terminer

7.2 Algorithmes d'échange inter-parcours

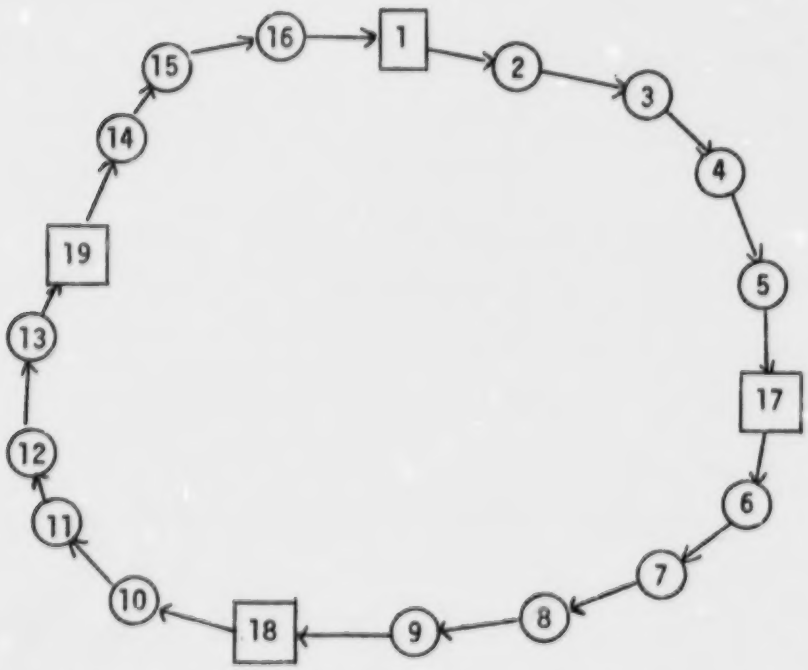
Le principe des améliorations inter-parcours est de permettre à la procédure OPT-2 des échanges d'arcs entre plus d'un parcours. Le cycle initial doit alors être constitué de tous les parcours reliés entre eux par des sommets duplicatats de l'école. La distance reliant l'école à ses duplicatats est assumée nulle pour permettre la réduction du nombre de parcours. Le changement majeur de la procédure OPT-2 vient du fait qu'il ne faille considérer que des échanges qui conservent la réalisibilité des parcours qui dépend de la distance, du nombre d'arrêts et de la capacité dans les autobus. Si ces trois critères sont respectés pour tous les parcours, alors nous procédons à l'échange d'arcs.

La figure 7.4 illustre comment un grand cycle est formé à partir d'une solution à plusieurs parcours. Dans notre implantation, l'entrée des sommets duplicatats de l'école dans notre réseau n'ajoute aucun arc. Le nombre de sommets initial étant connu, tout sommet additionnel portera un numéro supérieur. Lorsque nous demandons une distance dont l'un ou l'autre des numéros de sommets est supérieur au nombre initial, celui-ci est assumé être un duplicatat de l'école et la distance demandée est recherchée comme si celui-ci était l'école.

Fig. 7.4 Construction du cycle à partir de plusieurs parcours



Parcours originaux, 16 sommets



Cycle construit en reliant les parcours avec des sommets duplicatats d'école 17, 18 et 19

Nous devons discerner deux cas dans l'application des algorithmes d'échange d'arcs dans le grand cycle des parcours: le cas où le réseau est orienté et le cas où le réseau est non-orienté par rapport à l'école ou à ses duplicatats. Dans le cas où le réseau est orienté l'algorithme doit être modifié, ceci fait l'objet de la prochaine section. Nous pouvons par contre décrire les étapes du problème non-orienté lorsque la distance école-arrêt égale la distance arrêt-école. L'algorithme doit alors suivre la procédure décrite à la section 7.2.1.

7.2.1 Etapes de l'algorithme d'échange inter-parcours non-orienté

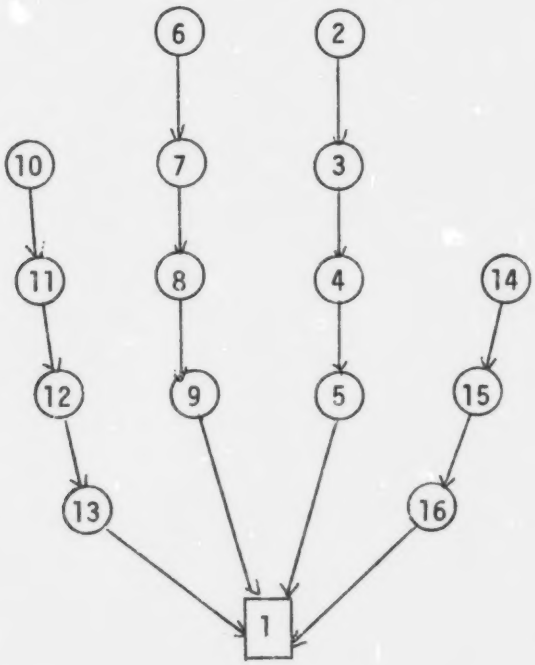
Etapes	Description
[A]	Créer un grand cycle avec tous les parcours en reliant les parcours avec des duplicatats d'école comme dans la figure 7.4
[B]	Appliquer une procédure OPT-2 pour trouver un échange qui réduirait la distance totale. . Si un échange est trouvé, faire l'étape [C] . Si aucun échange viable est trouvé, aller à l'étape [E]
[C]	Balayer tous les nouveaux parcours tel qu'ils seraient créés si l'échange serait réalisé. . si le nombre d'arrêts d'un parcours excède la contrainte fixée, aller à [D] . Si la distance d'un parcours excède la contrainte fixée, aller à [D] . Si la charge d'un parcours ne peut être alloué à un autobus libre, aller à [D] La nouvelle solution est réalisable, faire les échanges pour obtenir un nouveau cycle. Aller à l'étape [B]

Etapes	Description
[D]	La nouvelle solution n'est pas réalisable, retourner poursuivre les itérations de l'algorithme avec le parcours non-modifié
[E]	Remettre le grand cycle en parcours individuels
[F]	Terminer

7.3 Algorithme d'échange entre parcours sans boucles

Dans l'optique où nous traitons le problème des parcours sans boucles de départ à l'école, nous remarquons la structure d'arbre de deux-ci. En considérant l'école comme racine, chacun des parcours se rattache à celle-ci sous la forme d'une branche. La figure 7.5 illustre ceci.

Fig. 7.5



Les parcours sous forme d'arbre

La procédure d'échange des paires d'arcs OPT-2 peut modifier les parcours de deux façons. Elle peut réorganiser un parcours en modifiant la séquence de visite des arrêts. Elle peut aussi altérer deux parcours en échangeant des arrêts entre ceux-ci. Elle ne peut toutefois pas, dans aucune façon modifier plus de deux parcours.

Supposons maintenant que nous puissions conserver la propriété OPT-2 de chaque parcours. Ceci éliminerait les échanges à l'intérieur d'un seul parcours et ne laisserait donc que des échanges entre deux parcours. La propriété OPT-2 des parcours pourrait être conservée si après chaque échange entre deux parcours, une optimisation était faite pour chaque parcours avant de les enregistrer.

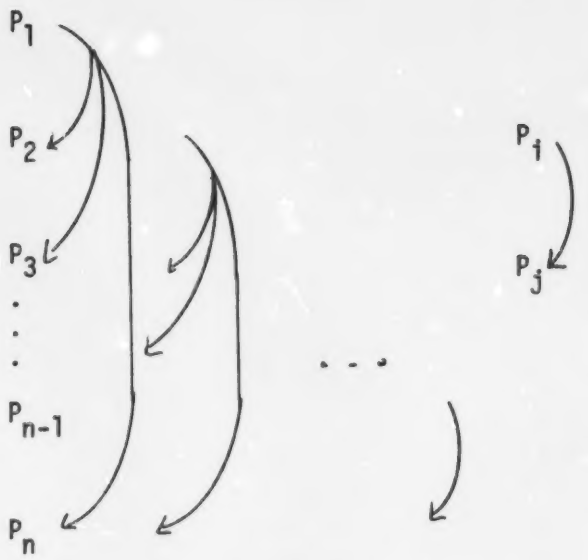
La réduction à un seul cas étant réalisable, nous pouvons maintenant décrire une heuristique d'échange entre les parcours. Cet heuristique peut même tenir compte d'essais infructueux à ne pas refaire. Supposons que nous ayons n parcours P_1, P_2, ..., P_n et une matrice Q, nxn, au départ à zéro. Cette matrice contiendra l'état des échanges effectués, un 0 si un échange n'a pas encore été tenté et un 1 si l'échange n'est pas réalisable. Les étapes de la section 7.3.1 et la figure 7.6 décrivent l'algorithme.

7.3.1 Etapes logiques de l'analyse des échanges entre parcours

Etapes	Description
[A]	Faire jusqu'à [D] pour i = 1, ... , (n-1)
[B]	Faire jusqu'à [D] pour j = (i+1), ... , n
	si Q_{ij} = 1, aller à [D]

Etapes	Description
[C]	Tenter un échange entre P_i et P_j
	. Si l'échange est réalisable, aller à [E]
	. Si l'échange n'est pas réalisable, poursuivre
[D]	Mettre 1 dans Q_{ij}
	Itérer ou Terminer
[E]	Exécuter l'échange entre P_i et P_j
	Optimiser P_i
	Optimiser P_j
	pour k = 1, ... , n $\quad Q_{ki} = Q_{i,k} = Q_{k,j} = Q_{j,k} = 0$
	recommencer à l'étape [A]

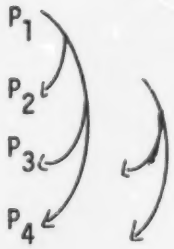
Fig. 7.6 Analyse des échanges entre parcours



(i)		P_1	P_2	(j) P_3	...	P_n
	P_1	0	0	0		0
	P_2	0	0	0		0
	P_3	0	0			0
	.					
	.					0
	.					
	P_n	0	0	0		0

Matrice d'essais Q

Exemple: Supposons quatre parcours, le seul échange possible est de P_3 à P_4, la matrice Q aura la forme suivante



Avant $P_3 \leftrightarrow P_4$

	P_1	P_2	P_3	P_4
P_1	0	1	1	1
P_2	0	0	1	1
P_3	0	0	0	0
P_4	0	0	0	0

P_1

P_2

P_3

P_4

Après ECHANGES
$P_3 \leftrightarrow P_4$

	P_1	P_2	P_3	P_4
P_1	0	1	0	0
P_2	0	0	0	0
P_3	0	0	0	0
P_4	0	0	0	0

La matrice Q indique (P_1,P_2) non échangeable mais (P_1,P_3), (P_1,P_4), (P_2,P_3), (P_2,P_4) sont maintenant permis puisque P_3 et P_4 ont été modifiés par l'échange. (P_3,P_4) demeure permis puisqu'il existe peut-être un second échange.

7.3.2 Echange entre les parcours P_i et P_j

Maintenant que nous avons décrit la première phase de l'algorithme, voyons la seconde: la procédure d'échange entre deux parcours P_i et P_j. Chacun des parcours enregistré en mémoire débute et se termine à l'école. Il s'agit donc de débuter sur un parcours pour ensuite terminer sur l'autre. Assumons les pointeurs I et J sur les parcours P_i et P_j respectivement ainsi que leurs successeurs S_I et S_J eux aussi sur le parcours. Nous devons remplacer les arcs (I,SI) et (J,SJ) par (I,SJ) et (J,SI) tel que présenté sur la figure 7.7. Si les parcours ainsi modifiés et par la suite optimisés demeurent réalisables et provoquent une réduction dans la distance totale de parcours, nous permettons leur réorganisation.

Il est évident que nous n'analysons pas tous les échanges réalisables possibles, nous limitant à ceux offrant un potentiel de réduction de distance.

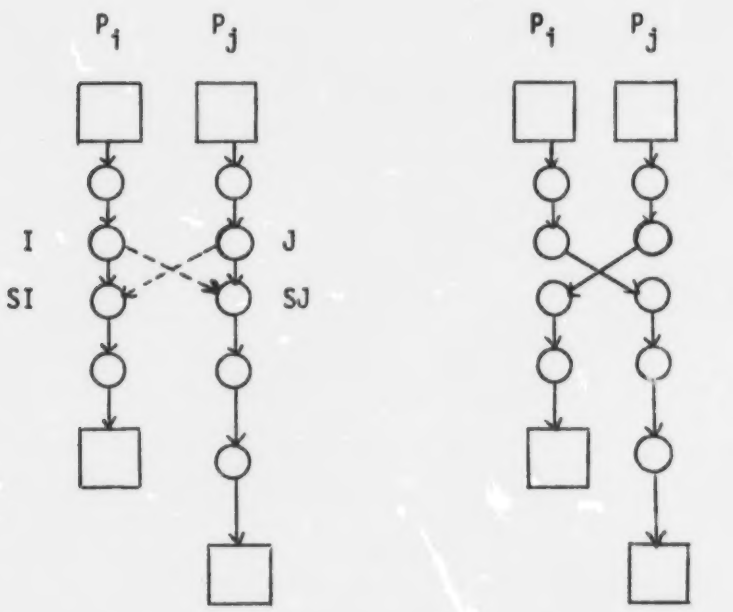
Si

$$d_{I,SJ} + d_{J,SI} \leq 1.1 \left[d_{I,SI} + d_{J,SJ} \right]$$

nous analyserons la réalisibilité de cet échange. Si celui-ci est réalisable mais ne diminue pas la distance totale après optimisation individuelle, nous conservons les parcours originaux. Il peut sembler utopique à première vue d'analyser des échanges dont le coût des nouveaux arcs est supérieur d'un facteur de 0.1 de celui des anciens arcs. Mais il faut considérer que chacun des parcours est par la suite optimisé. Il est de plus facile de comprendre ceci en considérant que l'échange crée deux nouveaux sous-ensembles d'arrêts avec lesquels nous devons former des parcours.

Notre algorithme OPT-2 d'échange entre parcours est décrit en entier dans la section suivante.

Fig. 7.7 Echanges analysés entre P_i et P_j



Parcours avant échange

Parcours après échange

7.3.3 Etapes logiques de l'algorithme d'échange entre parcours

Etapes	Description
[A]	Mettre Q_{ij} à 0 pour tous les n parcours $i = 1, \ldots, n \quad j = 1, \ldots, n$
[B]	Faire jusqu'à [P] pour $i = 1, \ldots, (n-1)$
[C]	Faire jusqu'à [O] pour $j = (i+1), \ldots, n$
[D]	Si $Q_{ij} \neq 0$, aller à [O].
[E]	Faire jusqu'à [M] variant I de l'école jusqu'au dernier arrêt de P_i Déterminer SI le successeur de I dans le parcours
[F]	Faire jusqu'à [L] variant J de l'école jusqu'au dernier arrêt de P_j Déterminer SJ le successeur de J dans le parcours
[G]	Si $(d_{I,SJ} + d_{J,SI}) > (d_{I,SI} + d_{J,SJ})$ 1.1 aller à [L]
[H]	Est-ce que les deux nouveaux parcours P'_i et P'_j ainsi formés respectent les contraintes de capacité d'autobus et de nombre d'arrêts. Sinon aller à [L]
[I]	Exécuter une procédure OPT-2 sur chacun des deux parcours P'_i et P'_j Si la distance de P'_i ou P'_j excède la contrainte de distance aller à [L]
[J]	Si la somme des distances des nouveaux parcours P'_i et P'_j est supérieure ou égale à la somme des distances des anciens parcours P_i et P_j, aller à [L]

Etapes	Description
[K]	Enregistrer les nouveaux parcours P'_i et P'_j
	Pour k = 1, ... , n
	$Q_{h,i} = Q_{i,k} = Q_{j,k} = Q_{k,j} = 0$
	pour permettre de nouveaux essais d'échange avec P'_i et P'_j
	Aller à l'étape [B]
[L]	Itérer sur les arrêts J du parcours P_j
[M]	Itérer sur les arrêts I du parcours P_i
[N]	Aucun échange possible entre P_i et P_j. mettre $Q_{ij} = 1$
[O]	Itérer j sur les parcours P_j
[P]	Itérer i sur les parcours P_i
[Q]	Il n'existe plus d'échanges de deux arcs possibles entre les parcours
	Terminer

8. Description des fichiers

Cette section définit les informations contenues dans les divers fichiers utilisés dans les étapes du traitement informatique de la génération des parcours. Nous ne ferons pas une liste exhaustive de la structure des enregistrements et des noms symboliques utilisés. Le lecteur intéressé pourra référer aux listes des instructions source du programme en annexe.

8.1 TAPE 1. Ecoles et arrêts

Le fichier TAPE 1 contient les informations de base pour générer les parcours. Ces données parviennent soit de l'étape informatique de l'assignation des étudiants aux arrêts, soit d'une assignation manuelle par le régisseur. Ces informations se présentent comme suit sur le fichier.

. Pour chaque traitement de génération de parcours

- . noeud de destination (1)
- . nombre d'arrêts
- . nombre d'écoles
- . pour chaque école
  - . code de bâtisse
  - . noeud de l'école
- . pour chaque arrêt
  - . noeud de l'arrêt
  - . nombre d'étudiants assignés

Lorsque le noeud de destination (1) est 0, toutes les écoles sont considérées comme des destinations. Le format de ce fichier est binaire, il peut contenir les données de plusieurs problèmes, les uns à la suite des autres sans indicateur de fin de fichier entre chacun.

8.2 TAPE 2. Données pour MINDIST

Le fichier TAPE 2 contient les informations nécessaires pour calculer les distances arrêts-écoles et inter-arrêt par le traitement MINDIST. Ce fichier est créé par le traitement INDIST qui utilise comme source d'informations le fichier TAPE 1. Le contenu du fichier se présente comme suit, où narr identifie le nombre d'arrêts et nbec le nombre d'écoles.

. Pour chaque problèmes de génération de parcours

. code de traitement	(4)	I
. nombre de noeuds à lire	(narr + nbec)	
. code d'écriture	(0)	
Pour chaque école (1, ..., nbec)		
. noeud de l'école . nombre de voisins	(narr + nbec - 1)	
Pour chaque arrêt (1, ..., narr)		
. noeud de l'arrêt 0		
. code de traitement	(4)	II
. nombre de noeuds à lire	(narr)	
. code d'écriture	(0)	
Pour chaque arrêt (1, ..., narr)		
. noeud de l'arrêt . nombre de voisins	(paramètre lu dans INDIST)	

La section I du fichier contient les informations requises pour le calcul des distances minimum entre les arrêts et chacune des écoles destinations, et des distances inter-écoles destinations. Dans cette section, chacun des noeuds doit apparaître une seule fois, une école destination ne peut donc pas être un arrêt. Dans la section II du fichier sont mises les informations requises pour le calcul des distances entre les arrêts. Pour chacun des arrêts, le nombre de voisins à atteindre avant de cesser le calcul des distances est spécifié. Ce paramètre est lu dans le programme INDIST et ne doit pas excéder le nombre maximum d'arrêts voisins (narr-1). Les noeuds des écoles et des arrêts sont respectivement en ordre ascendant dans le fichier et ne doivent pas se présenter deux fois dans une liste. Le format du fichier est de caractères numériques avec au moins un espace entre chaque chiffres.

8.3 TAPE 3. Distances minimum

Le fichier TAPE 3 contient les distances minimum calculées par le programme MINDIST. Ce programme n'étant pas de notre ressort, nous ne définirons que le format selon lequel nous l'utilisons.

. date du système (1)
. temps du système(1)

Pour chaque noeud pour lequel nous avons demandé au moins un voisin

. indice du noeud dans le réseau sur TAPE 8
. numéro du noeud considéré
. nombre maximum d'enregistrement à lire

Pour chaque noeud rencontré dans le calcul des distances

. numéro du noeud atteint
. numéro du noeud ancêtre dans le chemin
. distance du noeud de départ au noeud atteint

1 enregistrement de fin

. 0
. 0
. 0

(1) Ces informations proviennent du fichier Réseau sur TAPE 8

Le format des enregistrements est binaire. Les demandes formulées au programme MINDIST sont placées consécutivement sur ce fichier jusqu'à épuisement des données à traiter.

8.4 TAPE 7. Réseau des noeuds et segments

Le fichier TAPE 7 contient l'ensemble des noeuds et segments formant le réseau complet de transport permis. Ce fichier se présente sous le format ci-après décrit.

Pour chaque segment		Remarques
. code de municipalité		(1)
. code de rue		(1)
. code de segment		(1)
. libre		
. distance du segment		
. noeud origine		
. noeud destination		

. indicatif de fin de fichier (EOF)	

. nombre de noeuds	(NBNOD)	
. numéro de noeud	1, ..., NBNOD	(2)

. coordonnée X	1, ..., NBNOD
. coordonnée Y	1, ..., NBNOD

Remarque

1 Conservé en ordre croissant de code municipalité-rue-segment

2 Conservé en ordre croissant de numéro de noeud

8.5 TAPE 8. Réseau des noeuds et segments - structure interne

Le fichier TAPE 8 contient le réseau TAPE 7 sous la forme qu'il doit être conservé en mémoire(représentation sous forme d'échelle), sa description est donnée ci-après.

- date du système
- temps du système

- nombre d'écoles

Pour chaque école

- code de bâtisse
- indice du noeud

- nombre de noeud

Pour chaque noeud

- coordonnée X
- coordonnée Y
- numéro du noeud
- indice du premier segment incident au noeud
- code de bâtisse (0 si le noeud n'est pas une bâtisse)

- nombre de segments

Pour chaque segment

- indice du noeud d'origine
- indice du noeud destination
- pointeur au prochain segment incident au noeud origine
  (0 si dernier dans la liste)
- pointeur au prochain segment incident au noeud destination
  (0 si dernier dans la liste)
- distance du segment
- code de municipalité
- code de rue et de segment concatenné

8.6 TAPE 9. Réseau des arrêts

Le fichier TAPE 9 contient les informations sur les écoles et arrêts prêtes à être utilisées par le traitement de génération des parcours ROUTE. Il est généré dans INROUT à partir d'informations contenues sur TAPE 3, TAPE 11 et TAPE 7. Le fichier se présente sous le format suivant.

Pour chaque problème à traiter

- . nombre d'écoles concernées (nbec)

Pour chaque bâtisse 1, ..., nbec

- . code de bâtisse
- . noeud de la bâtisse

- . nombre de destinations nbd

Pour chaque destination I = 1, ..., nbd - Patrons de visite

- . noeud destination (I), séquence de visite des (nbd-1) autre noeuds de destination partant du noeud destination (I)

- . noeud de l'école destination
- . 0
- . nombre d'arrêts
- . coordonnée X
- . coordonnée Y

narr

Pour chaque arrêt 1, ..., narr

- noeud de l'arrêt
- code de noeud artificiel
- nombre d'étudiants assignés
- coordonnée X
- coordonnée Y
- distance partant de l'arrêt pour visiter toutes les destinations
- numéro du patron de visite des destinations pour l'arrêt

Pour chaque arrêt et pour autant de voisins demandés

- noeud de l'arrêt i
- code de noeud artificiel i
- noeud de l'arrêt j
- code de noeud artificiel j
- distance entre les arrêts i et j

Enregistrement de fin de problème

- 0
- 0
- 0
- 0
- 0

Notons que le noeud de l'école destination est le premier noeud des destinations du problème. Les heuristiques du programme ROUTE requièrent la présence d'un seul noeud destination; ce noeud est aussi utilisé pour localiser la fin des parcours dans un tableau représentant les parcours générés[1]. Notons aussi qu'un arrêt est identifié de façon unique par

1 Le noeud de destination peut être différent des noeuds des bâtisses dans le cas d'un point de transfert par exemple. Les codes et noeuds de bâtisses ne servent donc qu'à titre d'information.

ses numéros de noeud et de noeud artificiel. L'arrêt est sectionné en arrêt artificiel par INROUT lorsque celui-ci excède un maximum d'étudiants spécifié. Le fichier peut contenir le réseau des arrêts pour plusieurs problèmes, les uns à la suite des autres.

8.7 TAPE 10 . Parcours

Le fichier TAPE 10 contient les parcours de la meilleure solution de chaque problème traité par le programme ROUTE. A chaque fois que nous écrivons sur ce fichier, nous ajoutons des parcours à ceux déjà existant au fichier. Sa dimension s'accroît donc à chaque traitement. Le format de ce fichier est binaire et se présente comme suit:

Pour chaque problème traité

Pour chaque parcours

- numéro du parcours
- nombre de destination nbd
- 1^e destination visitée

≷

- nbd^e destination visitée
- 1^e code de transfert = 0

≷

- nbd^e code de transfert = 0
- nombre d'arrêts dans le parcours nba
- 1^e noeud de l'arrêt (premier arrêt)

≷

- nba^e arrêt (dernier arrêt)
- distance du parcours

8.8 TAPE 11. Fichier intermédiaire écoles et arrêts

Le fichier TAPE 11 est utilisé pour passer les informations du programme INDIST au programme INROUT, afin que ce dernier puisse reconnaître les calculs de distance demandés au programme MINDIST et générer le fichier TAPE 9. Le format du fichier est binaire et se présente comme suit:

Pour chaque problèmes à traiter

- . nombre de bâtisses — nbec

Pour chaque bâtisse (1, ..., nbec)

- . code de bâtisse
- . noeud de la bâtisse

- . nombre de destinations — nbd

Pour chaque destination (1, ..., nbd)

- . noeud de l'école destination

- . nombre d'arrêts — nba

Pour chaque arrêt

- . noeud de l'arrêt
- . nombre d'étudiants

9. Les étapes informatiques de la génération des parcours

9.1 Création manuelle du fichier des arrêts

L'étape de création du fichier des arrêts est optionnelle dans le traitement. Elle est utilisée lorsqu'un régisseur désire déterminer lui-même ses arrêts. Ceci survient généralement lorsque le fichier généré par l'assignation informatique des arrêts est intercepté et modifié par le régisseur.

9.1.1 Diagramme des entrées-sorties

9.1.2 But du programme

Le programme ALIRE lit les arrêts et écoles requises à la génération des parcours sous un format libre de cartes et crée avec ces données un fichier TAPE 1 semblable à celui généré par l'étape informatique de l'assignation des étudiants aux arrêts.

9.1.3 Description des entrées

Les entrées du programme sont sur cartes ou images de cartes, en format libre. C'est à dire que les données n'apparaissent pas à des colonnes spécifiques, elles doivent être séparées par une ou plusieurs espaces. Les données requises sont présentées dans le tableau 9.1. Plusieurs problèmes peuvent être donnés séquentiellement dans les cartes d'entrée en omettant la date et le temps du système pour les données consécutives au premier problème.

Tableau 9.1

Informatique	Description	Remarque
- date du système		1
- temps du système		1
Pour chaque problème		
- noeud destination		2
- nombre d'arrêts	narr	
- nombre de bâtisses	nbec	3
- code de bâtisse	pour chaque bâtisse 1,...,nbec	
- noeud de la bâtisse		
- noeud de l'arrêt	pour chaque arrêt 1,...,narr	4
- nombre d'étudiants		

Remarque

1 La date et le temps du système tel que généré lors de la création du fichier RESEAU (TAPE 7). Cette donnée ne cause pas de rejet lorsqu'incompatible. Cette information ne doit apparaître qu'une fois au début des cartes.

2 Lorsque le noeud destination est à zéro, les noeuds de bâtisses sont les destinations. Si le noeud destination est présent, celui-ci peut ne pas être un noeud de bâtisse (un point de transfert par exemple).

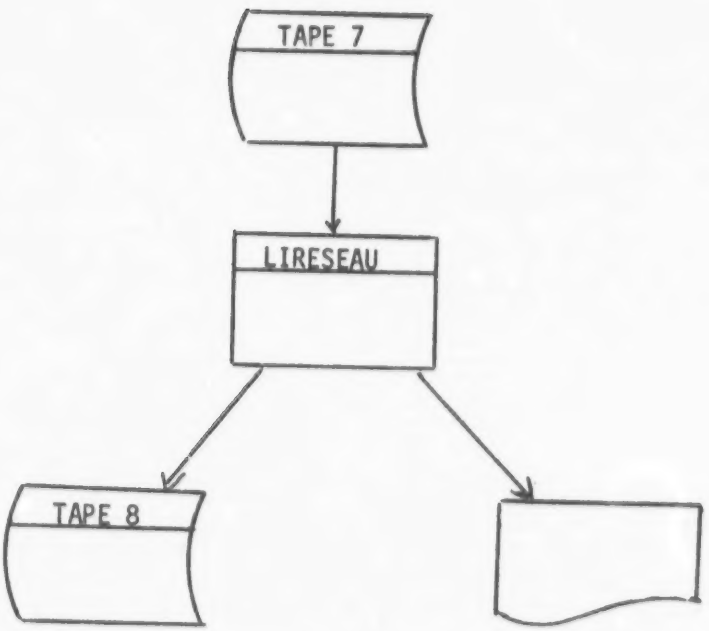
3 Au moins une bâtisse doit être présente.

4 Les noeuds d'arrêts qui sont des noeuds destinations sont rejetés. Les noeuds d'arrêts sans étudiants sont rejetés.

9.2 Création du réseau des noeuds et segments structure interne

L'étape de création du fichier TAPE 8 contenant le réseau des noeuds et segments sous une structure de représentation interne est essentielle à l'exécution du calcul des distances (section 9.4). Ce programme est extrait du travail de G. Dulac [D2] où il y est documenté. Le lecteur intéressé pourra donc référer à ce texte pour une description des entrées-sorties et du programme.

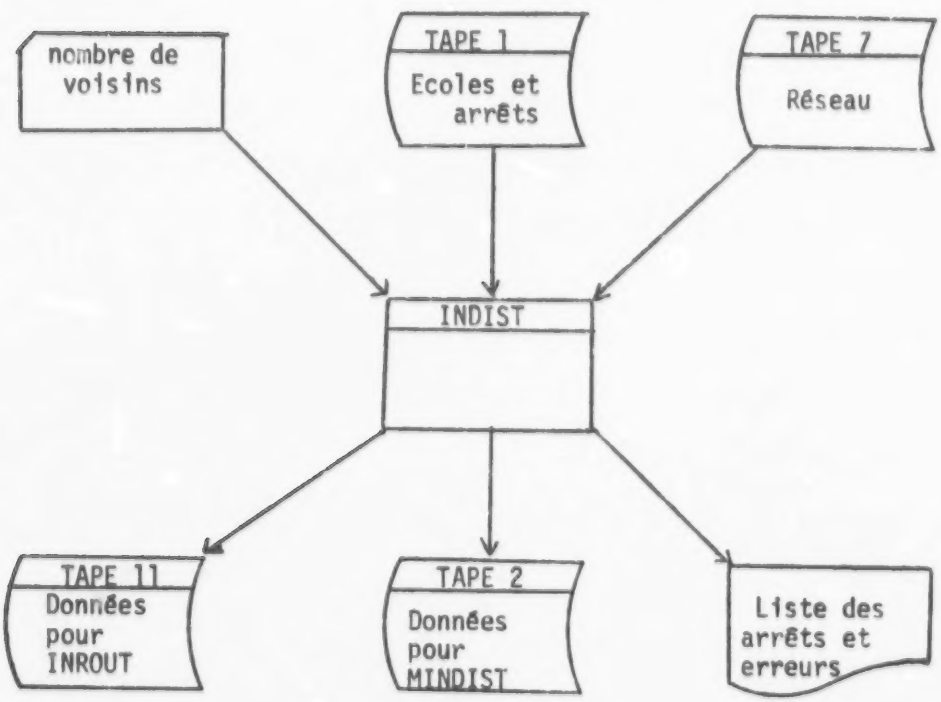
9.2.1 Diagramme des Entrées-Sorties



9.3 Génération des demandes de calcul des distances

L'étape de la génération des demandes de calcul des distances est essentielle au traitement de génération des parcours dans le sens qu'elle prépare à partir des destinations et arrêts du fichier TAPE 1, une liste de requêtes pour le calcul des distances par l'étape MINDIST. Outre ceci, un fichier intermédiaire TAPE 11 est aussi créé pour passer l'information à l'étape INROUT. Le régisseur fournit sur une carte le nombre d'arcs inter-arrêt à calculer par arrêt (normalement ce nombre devrait être autour de dix) ce qui spécifie alors à MINDIST d'arrêter les itérations pour chaque arrêt lorsque la distance des dix arrêts les plus rapprochés est connu. L'étape peut traiter plusieurs problèmes consécutifs sur le fichier TAPE 1.

9.3.1 Diagramme des entrées-sorties



9.3.2 Etapes logiques du programme INDIST

Etapes	Description
[A]	Lire les noeuds du réseau sur le fichier TAPE 7 pour valider les données
[B]	Lire la carte spécifiant le nombre de voisins (NV), si la carte est manquante assumer 10 voisins
[C]	Lire sur TAPE 1 le noeud destination, les noeuds de bâtisse, les noeuds des arrêts et le nombre d'étudiants. Si un noeud n'existe pas dans la liste des noeuds du réseau, terminer la production. Si un noeud d'arrêt est aussi un noeud destination, rejeter l'arrêt. Si un arrêt n'a aucun étudiant d'assigné, rejeter l'arrêt.
[D]	Définir la ou les destinations. Créer les destinations en ordre ascendant de numéro de noeud. Créer les arrêts en ordre ascendant de numéro de noeud.
[E]	Générer sur TAPE 2 une première requête pour calculer les distances inter-destinations et arrêts-destinations
[F]	Générer sur TAPE 2 une seconde requête pour calculer pour chaque arrêt la distance des arcs des (NV) arrêts les plus rapprochés
[G]	Ecrire sur TAPE 11 la liste des destinations, bâtisses, arrêts et nombre d'étudiants traités
[H]	Aller à l'étape [C] si un autre problème est demande, sinon terminer

9.3.3 Description des entrées

L'usager fournit une carte ou image de carte au programme pour lui indiquer le nombre d'arcs inter-arrêts à calculer par arrêt. Le format est libre mais normalement le nombre devrait être cadré à gauche. Si la carte est absente, le nombre est par défaut 10. Si le nombre d'arcs à calculer excède le maximum possible, il est automatiquement réajusté au dit maximum.

9.3.4 Description des sorties

Pour chacun des problèmes traités l'usager reçoit une liste explicative de ses données. La liste contient le nombre et la liste des bâtisses et des noeuds associés, le noeud de destination et le nombre d'arrêts spécifiés, le noeud de chaque arrêt et le nombre d'étudiants assignés. Les noeuds rejetés sont indiqués par un message. A la fin de la liste le nombre d'arrêts retenus et le nombre de voisins sont indiqués. Un exemple de cette liste est présenté ci-après.

*** SYSTEME DE TRANSPORT SCOLAIRE ***

*** PREPARATIFS DU CALCUL DES DISTANCES ***

NOMBRE DE BATISSES: 3

BATISSE	NOEUD
48	163913050
71	163702237
60	174985778

NOEUD DE L'ECOLE: 0 NOMBRE D'ARRETS: 85

NOEUD DE L'ARRET	NOMBRE D'ETUDIANTS
143816437	2
153451693	1
153713807	9
153918458	16
154132585	7
154440220	8
154636074	12
151922318	1
151952372	4
151955506	1
152070665	8
152145528	2
152218798	3
161578293	3
162061568	7
162013105	1
162948129	3
163392362	1
163510152	1
153543978	1
153239160	2
153190765	7
153507409	7
153652160	4
163360637	4
163260578	1
163702207	2 REJET, CE NOEUD EST UNE ECOLE
164192318	1
174086071	20
183024896	15
173150375	2
173055300	5
172738573	3
172535745	2
163976020	18
170216240	11
174315364	23
174840542	17
174804709	21
184454058	12
174537081	22
174683267	14
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	4
[illegible]	3
[illegible]	4
[illegible]	[illegible]
[illegible]	[illegible]

180234662	3	
183982802	4	
165118838	21	
165721372	8	
165608220	3	
165368101	12	
165252250	5	
165965963	11	
175300074	19	
175240708	11	
175046858	18	
174703823	12	
174555941	8	
186037695	0	REJET PAS D'ETUDIANTS
186230957	11	
170212120	8	
170013412	8	
175872614	5	
175746990	2	
175622989	4	
175510630	11	
197473467	8	
197521112	3	
16759971	4	
16777988	4	
197926140	5	
197646331	1	
197469498	0	REJET PAS D'ETUDIANTS
187443044	2	
187262037	6	
187185180	7	
180954411	5	
180680559	7	
186442632	7	
186037695	10	
GENERE POUR ECOLE NOEUDS	0	82 ARRETS AVEC 15 VOISINS

*** SYSTEME DE TRANSPORT SCOLAIRE ***

--- PREPARATIFS DU CALCUL DES DISTANCES ---

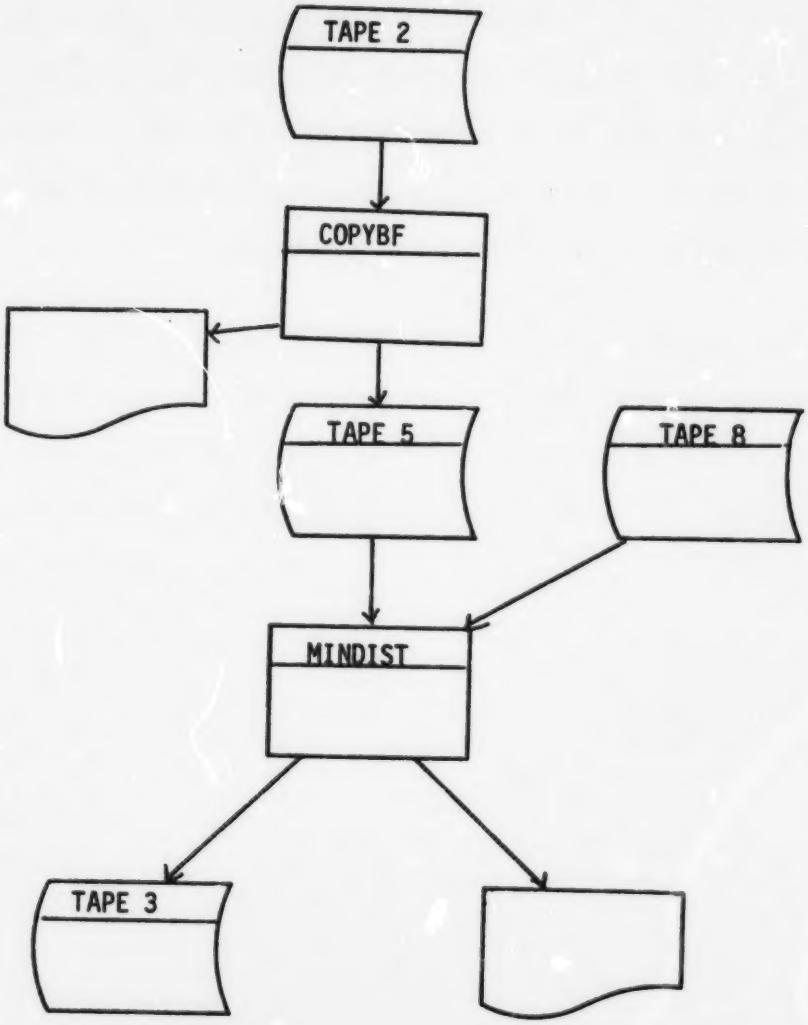
FIN NORMALE, LU 1 ECOLES,GENERE 1 ECOLES

9.4 Calcul des distances inter-noeuds

L'étape du calcul des distances inter-noeuds calcule la distance des arcs qui relient deux noeuds. Un arc est constitué d'un ensemble de segments identifiant le chemin qui doit être parcouru pour joindre les deux noeuds. Le programme de calcul des distances MINDIST est une variante de l'algorithme de DIJKSTRA, et il est extrait du travail de G. Dulac [D2] où il y est documenté. Le lecteur intéressé pourra référer à ce texte pour une description des entrées-sorties et du programme.

Dans notre implantation nous avons modifié les sorties du programme pour éliminer les noeuds intermédiaires dans le calcul des distances entre deux noeuds, et nous avons remplacé l'indice du noeud par le numéro du noeud. Le fichier TAPE 3 décrit le nouveau format du fichier.

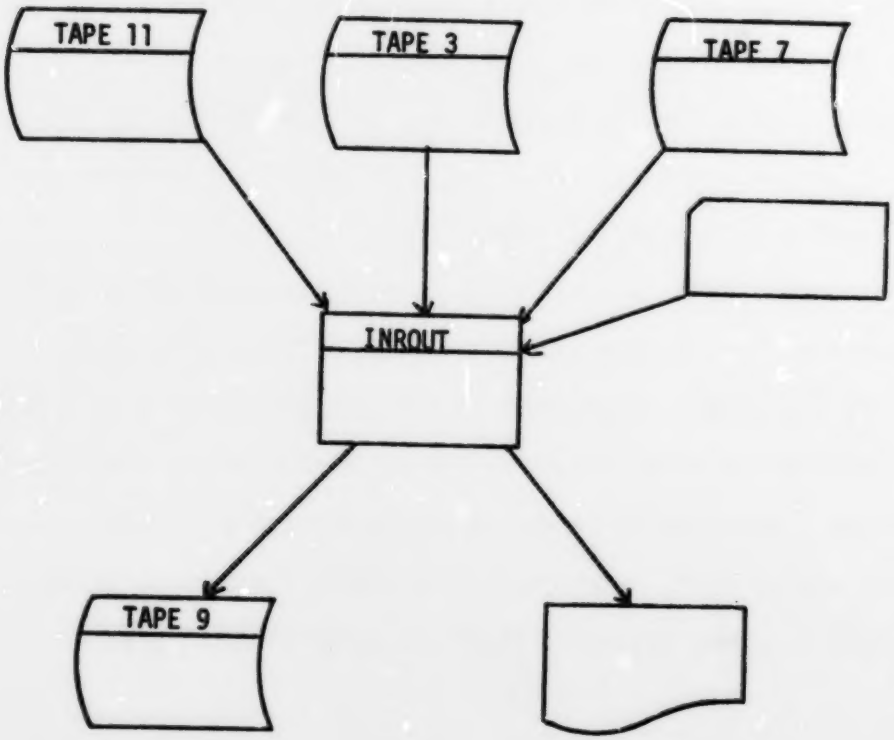
9.4.1 Diagramme des entrées-sorties



9.5 Création du réseau école-arrêts

L'étape de la création du réseau école-arrêts fusionne les informations de différents fichiers pour enfin créer les données nécessaires à la génération des parcours. Le réseau école-arrêts créé contient la définition des bâtisses traitées, les patrons identifiant la séquence de visite des destinations, le noeud représentant l'école, les arrêts et étudiants du réseau et l'ensemble des arcs et distances inter-arrêts calculées. L'usager fournit une carte pour spécifier le nombre d'étudiants maximum par arrêt et le nombre maximum d'arcs inter-arrêts à générer par arrêt. L'étape peut traiter plusieurs problèmes consécutifs sur le fichier TAPE 11.

9.5.1 Diagramme des entrées-sorties



9.5.2 Etapes logiques du programme INROUT

Etapes	Description
[A]	Lire les noeuds, coordonnées X et Y du réseau sur le fichier TAPE 7 (pour générer les coordonnées de l'école et des arrêts)
[B]	Lire la carte spécifiant le nombre maximum d'étudiants par arrêt et le nombre d'arcs inter-arrêts à générer par arrêt. Si la carte est absente assumer 15 étudiants par arrêts et 10 arcs inter-arrêts
[C]	Lire les bâtisses, les destinations, l'école destination, les arrêts et le nombre d'étudiants sur le fichier TAPE 11 . Si on a épuisé le fichier TAPE 11 , terminer
[D]	Lire les distances inter-destinations et les distances arrêts-destinations sur le fichier TAPE 3
[E]	Calculer les patrons de visite des destinations avec l'algorithme OPT-2 modifié pour un quasi-tour, c'est-à-dire partant de chaque destination, la distance requise pour visiter les autres destinations (sans revenir au point de départ). Conserver ces patrons de visite et leur coût
[F]	Pour chaque arrêt, choisir la première destination où se rendre pour minimiser le tour des destinations. Calculer le coût arrêt-destination comme étant la distance de l'arrêt à la première destination plus le coût de visite partant de cette même destination. Conserver le patron de visite associé à l'arrêt.
[G]	Pour chaque arrêt, déterminer si le nombre d'étudiants assignés excède le nombre maximum d'étudiants spécifié par l'usager.

Etapes	Description
	Si oui - déterminer le nombre de duplicatats requis. - distribuer uniformément les étudiants parmi l'arrêt original et ses duplicatats
[H]	Pour tous les arrêts duplicatés - générer des distances de 1 mètres entre l'arrêt original et ses duplicatats et entre les duplicatats
[I]	Pour chaque arrêt, associer les coordonnées X-Y
[J]	Ecrire sur le fichier TAPE 9, le nombre de bâtisses, les bâtisses et leurs noeuds, le nombre de patron de visite des destinations et les patrons de visite, le noeud de l'école destination et ses coordonnées X-Y, le nombre d'arrêts, les noeuds des arrêts et leurs code de duplicatat, leur nombre d'étudiants, leurs coordonnées X-Y, leurs distance arrêt-destinations et leurs codes de patron de visite
[K]	Pour chaque arrêt original - lire sur le fichier TAPE 3 les distances inter-arrêts calculées - écrire sur le fichier TAPE 9, jusqu'à concurrence du nombre d'arcs par arrêt spécifié par l'usager, 1^{o} les arcs et distances inter-arrêt calculés par MINDIS7 2^{o} si un des deux arrêts ou les deux possèdent des duplicatats, générer les arcs et distances pour chaque duplicatat d'arrêts de la même façon qu'ils l'ont été pour les arrêts originaux
[L]	Aller à l'étape [C]

9.5.3 Description des entrées

L'usager fournit une carte ou image de carte au programme pour lui indiquer le nombre maximum d'étudiants par arrêt et le nombre maximum d'arcs inter-arrêt à générer pour chaque arrêt. Ces données sont fournies en format libre, c'est-à-dire que les deux nombres doivent être séparés par au moins une espace. Le format d'entrée est décrit dans le tableau 9.5. Lorsque l'usager omet cette carte, 15 étudiants par arrêts et 10 arcs inter-arrêt sont assumés par défaut.

Tableau 9.5

Séquence d'entrée	Remarque
- 0 ou plusieurs espaces	
- nombre maximum d'étudiants	(1)
- 1 ou plusieurs espaces	
- nombre maximum d'arcs inter-arrêt	

Remarque

(1) Si le nombre maximum d'étudiants est inférieur à 1, la valeur par défaut de 15 est assumée.

9.5.4 Description des sorties

Pour chacun des réseaux générés l'usager reçoit une liste explicative du contenu du fichier TAPE 9 généré. La liste contient le nombre maximum d'étudiants par arrêts et le nombre de voisins par arrêts tels que spécifiés par l'usager. La liste des bâtisses concernées, le nombre et les patrons de visite des destinations, la description du noeud destination considéré. Pour chaque arrêt nous avons le numéro de noeud, son code de

duplicatat, le nombre d'étudiants assignés, les coordonnées X-Y, la distance de l'arrêt aux destinations et le patron de visite des destinations. Le nombre d'arcs inter-arrêt généré est donné après la liste des arrêts. Il est à noter que ce dernier est le nombre maximum d'arcs générés, en fait si nous avons l'arc (i,j), il est très probable d'avoir plus loin l'arc (j,i). Or, ceux-ci sont considérés comme un seul arc dans le réseau des arrêts. Un exemple de cette liste est présenté ci-après.

*** SYSTEME DE TRANSPORT SCOLAIRE ***

--- CREATION DU RESEAU DES ARRETS ---

MAXIMUM D'ETUDIANTS PAR ARRET: 15
MAXIMUM DE VOISINS PAR ARRET: 10

NOUS TRAITONS 3 ECOLES

BATISSE	NOEUD
71	163702207
44	163913050
60	174985778

NOUS AVONS 3 PATRONS DE VISITE DES DESTINATIONS

NO DU PATRON	SEQUENCE DES NOEUDS		
1	163702207	163913050	174985778
2	163913050	163702207	174985778
3	174985778	163913050	163702207

NOEUD DESTINATION: 163702207 NB.ARRETS: 94 X: 693702 Y: 5083207

ARRET	#DUPL.	NR.ETUD.	X	Y	DIST.	#PATRON
16759971	0	4	697585	5079713	6111	3
16777948	0	4	697770	5079483	6406	3
141578293	0	3	691578	5085293	5023	1
142041548	0	7	692061	5085548	5177	1
142613105	0	1	692613	5085105	4471	1
142948129	0	3	692948	5085129	4450	1
143239160	0	2	693239	5085100	4451	1
[illegible]	0	1	693392	5085362	4704	1
143510152	0	1	693514	5085152	4906	1
143816037	0	2	693816	5085037	4462	1
151852372	0	4	691852	5084372	4790	1
151922318	0	1	691922	5084318	4702	1
151955506	0	1	691955	5084506	4673	1
152070665	0	8	692070	5084665	4477	1
152145628	0	2	692145	5084628	4394	1
[illegible]	0	3	692218	5084794	4218	1
153190765	0	7	[illegible]	5084765	3692	1
153343978	0	1	693343	5084978	4660	1
153567069	0	7	693567	5084469	3816	1
153652164	0	4	693652	5084164	2990	1
153713807	0	6	693713	5084807	3772	1
153851802	0	7	693851	5084802	3918	1
153914454	0	9	693914	5084452	3307	1
[illegible]	1	7	693914	5084454	3307	1
154132585	0	7	694132	5084585	3534	1
154440220	0	8	694440	5084220	2975	1
154036074	0	12	694036	5084074	3185	1
163076020	0	10	693076	5083020	2031	1
163076020	1	8	693076	5083020	2031	1
163296578	0	1	693296	5083578	2328	1
163468837	0	4	693468	5083837	2646	1
164192418	0	1	694192	5083418	2438	1
165065013	0	11	695065	5083013	2313	3
165110434	0	11	695110	5083434	3142	3
165110434	1	10	695110	5083434	3142	3
165254256	0	5	695254	5083256	2311	3
165368431	0	12	695368	5083431	2499	3
165638220	0	3	695638	5083220	2799	3
165721372	0	6	695721	5083372	2982	3
172734373	0	3	692738	5062373	3458	1

172836705	0	2	692836	5082705	3026	1
173055340	0	7	693055	5082340	3510	1
173150075	0	2	693150	5082075	3345	1
174068071	0	11	694068	5082071	3091	3
174068071	1	9	694068	5082071	3091	3
174216246	0	11	694216	5082246	2863	3
174315364	0	12	694315	5082364	2710	3
174315364	1	11	694315	5082364	2710	3
174464542	0	9	694464	5082542	2479	3
174464542	1	8	694464	5082542	2479	3
174547081	0	12	694547	5082081	2649	3
174547081	1	10	694547	5082081	2649	3
174555941	0	6	694555	5082941	2301	3
174604709	0	11	694604	5082709	2262	3
174604709	1	10	694604	5082709	2262	3
174683267	0	9	694683	5082267	2420	3
174683267	1	7	694683	5082267	2420	3
174703823	0	12	694703	5082823	2112	3
174816445	0	13	694816	5082445	2199	3
175048858	0	10	695048	5082858	1825	3
175048858	1	8	695048	5082858	1825	3
175244708	0	11	695244	5082708	2066	3
175360674	0	10	695360	5082674	2176	3
175360674	1	9	695360	5082674	2176	3
175510830	0	11	695510	5082830	2396	3
175622980	0	4	695622	5082980	2582	3
175746870	0	2	695746	5082870	2733	3
175872614	0	5	695872	5082614	2913	3
176013412	0	8	696013	5082412	3165	3
176212124	0	8	696212	5082124	3521	3
183924896	0	15	693924	5081896	3316	3
183982842	0	[illegible]	693982	5081842	3395	3
184234662	0	3	694234	5081662	3169	3
184371560	0	14	694371	5081560	3217	3
184454958	0	12	694454	5081958	2402	3
184759268	0	8	694759	5081268	3722	3
184885170	0	8	694885	5081170	3609	3
185098022	0	8	695098	5081022	4117	3
185368102	0	8	695368	5081102	4109	3
185658494	0	9	695658	5081494	3617	3
185658494	1	7	695658	5081494	3617	3
186037695	0	10	696037	5081695	3529	3
186230957	0	11	696230	5081957	3677	3
186442632	0	7	696442	5081632	4075	3
186640559	0	7	696640	5081559	4323	3
186954411	0	5	696954	5081411	4636	3
187185140	0	7	697185	5081140	4991	3
187293037	0	6	697293	5081037	5132	3
187443044	0	2	697443	5081044	5262	3
185232917	0	3	695232	5080917	4296	3
187473067	0	8	697473	5080067	5352	1
187521112	0	3	697521	5080112	5708	3
187646341	0	1	697646	5080341	5500	3
187926140	0	5	697926	5080140	5909	3

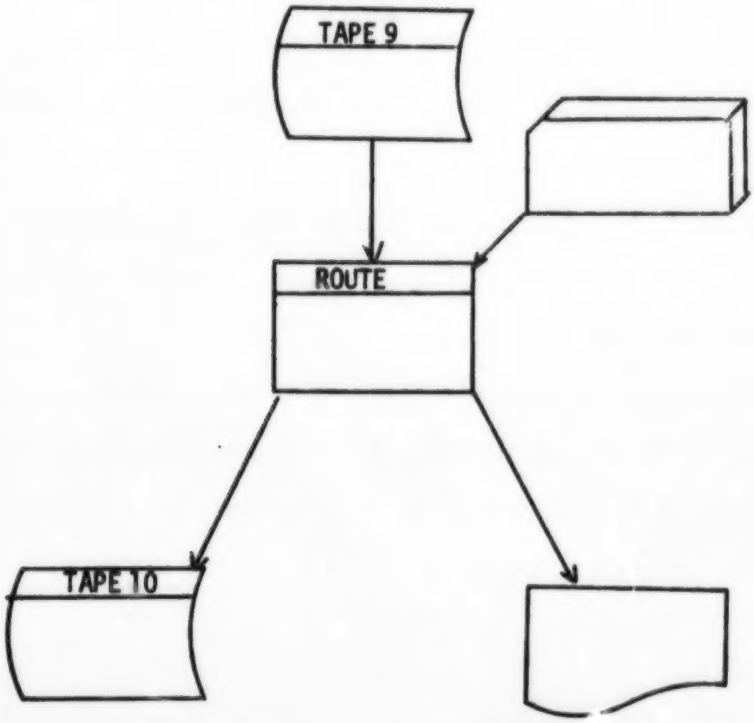
NOUS AVONS GENERE 1111 ARCS INTER-ARRETS.

```
*** SYSTEME DE TRANSPORT SCOLAIRE ***
*** CREATION DU RESEAU DES ARRETS ***
FIN NORMALE
```

9.6 Génération des parcours

L'étape de la génération des parcours construit des parcours respectant les contraintes de capacité, de nombre d'arrêts et de distance de parcours avec les heuristiques décrits dans les sections précédentes. L'usager fournit par paramètres les contraintes et heuristiques demandés pour chaque problème. Le programme se charge alors de créer des parcours réalisables, de choisir et d'écrire la meilleure solution sur le fichier TAPE 10 et d'imprimer les solutions à distance minimum pour chaque nombre de parcours différents.

9.6.1 Diagramme des Entrées-Sorties



9.6.2 Etapes logiques du programme ROUTE

Etapes	Description
[A]	- Lire les contraintes de distance, de nombre d'arrêts, les types d'autobus disponibles, le nombre et la capacité pour chacun des types. - Si aucune données est disponible, Terminer
[B]	- Lire le réseau des arrêts sur TAPE 9 et le conserver en mémoire sous la représentation interne d'échelle. - Si on a épuisé le fichier TAPE 9 avant la fin normale, Terminer
[C]	Lire une carte de l'usager indiquant l'heuristique, le type de formule, le type d'optimisation de parcours, le type d'échange inter-parcours, le code avec ou sans origine des parcours à l'école et la description du paramètre λ lorsque requis
[D]	Valider la carte lue en cas d'erreur, rejeter la demande et aller à l'étape [C] . Si la carte est la carte de fin des demandes pour le problème, aller à l'étape [M]
[E]	Faire jusqu'à l'étape [L] pour chaque valeur du paramètre λ demandés dans la carte. Si le paramètre λ n'est pas spécifié, faire une fois pour $\lambda = 0.0$
[F]	Si l'heuristique demandé est Clarke & Wright, générer les parcours selon les étapes décrites à la section 4.1.4 avec le type de formule (4.5)' ou (4.11)' et le paramètre λ spécifié par l'usager. Aller à l'étape [K]

Etapes	Description
[G]	Si l'heuristique demandé est Gillet & Miller, générer les parcours selon les étapes décrites à la section (5.1.1), aller à l'étape [K]
[H]	Si l'heuristique demandé est Newton & Thomas, générer les parcours selon les étapes décrites à la section (5.2.2), aller à l'étape [K]
[I]	Si l'heuristique demandé est INSERTION, générer les parcours selon les étapes décrites à la section (6.3) avec le type de formule (6.2) ou (6.3) et le paramètre λ spécifié pour la formule (6.3) par l'usager, Aller à l'étape [K]
[J]	Si l'heuristique demandé est Yellow, générer les parcours selon les étapes décrites à la section (4.2.2) avec le type de formule (4.5)' ou (4.11)' et le paramètre λ spécifié par l'usager. Aller à l'étape [K]
[K]	Selon le nombre de parcours généré par l'heuristique demandé, comparer la distance totale des parcours avec l'ancienne solution retenue pour le même nombre de parcours. Si la distance totale est inférieure ou s'il n'existe pas d'autres solutions avec le même nombre de parcours, conserver les paramètres requis pour obtenir la solution actuelle, et la distance totale de parcours
[L]	Itérer sur les paramètres λ demandés par l'usager, si nous avons épuisé la liste des paramètres demandés, aller à l'étape [C]
[M]	Trouver parmi la liste des solutions retenues, celle qui minimise le facteur parcours-distance

Etapes	Description
[N]	Pour chacune des solutions retenues faire jusqu'à l'étape [R] . Refaire les étapes [F] à [J] avec les paramètres conservés pour réobtenir la même solution. Si on a épuisé la liste des solutions retenues, aller à l'étape [A]
[O]	Si l'amélioration inter-parcours est demandé, faire l'algorithme décrit à la section 7.2.1 si l'on traite le problème avec départ et arrivée à l'école ou l'algorithme décrit à la section 7.3.3 si l'on traite le cas sans origine déterminée
[P]	Si cette solution est la meilleure telle que déterminée dans l'étape [M] , écrire cette solution sur le fichier TAPE 10
[Q]	Imprimer la solution
[R]	Créer à l'aide des coordonnées X-Y des arrêts une matrice géographique décrivant les parcours. Imprimer cette matrice. Aller à l'étape [N]

9.6.3 Description des entrées

L'usager doit fournir pour chacun des problèmes à traiter les contraintes et heuristiques demandés sur cartes ou images de cartes. A la fin de ses demandes pour un problème, il doit placer une carte de fin de demande. La séquence et le contenu des cartes pour un problème est décrit dans le tableau ci-après.

Tableau 9.6

Information	Colonne	Longueur	Remarque
Titre (nom de l'organisme)	1	80	
Pour chaque problème à traiter			
- nombre de types d'autobus (nbus)	1	2	(2)
- nombre maximum d'arrêts	6	5	
- distance maximum	11	5	
- numéro du premier parcours	16	5	(1)
Pour chaque type d'autobus 1,...,nbus			
- nombre d'autobus disponibles	1	5	(2)
- capacité de l'autobus	6	5	(2)
Pour chaque heuristique désiré			
- code d'heuristique	1	1	(3)
- code de formule	2	1	(4)
- code d'optimisation	3	1	(5)
- code d'échange inter-parcours	4	1	(6)
- code d'origine des parcours	5	1	(7)
- λ initial (format F5.3)	6	5	(8)
- λ final (format F5.3)	11	5	(8)
- incrément de λ (format F5.3)	16	5	(8)
A la fin des heuristiques			
- code d'heuristique à zéro	1	1	(9)

Remarques	Description
(1)	Les parcours enregistrés aussi bien que ceux imprimés pour l'usager sont numérotés à partir de ce nombre par incrément de 1.
(2)	L'usager peut spécifier jusqu'à 10 types d'autobus, pour chacun il doit en donner le nombre disponible et la capacité. (Les algorithmes Clarke & Wright et Yellow fonctionnent par réduction du nombre de parcours, au départ un autobus de type 1 est assigné à chaque arrêt, le nombre d'autobus utilisé peut donc être supérieur au nombre d'autobus disponibles et empêcher la réduction du nombre de parcours. On suggère à l'usager de réserver le type 1 pour des autobus artificiels de capacité 0 avec une grande disponibilité. Sitôt que deux arrêts seront regroupés sous un même parcours, l'algorithme assignera un autobus d'un autre type pour respecter la contrainte de capacité.)
(3)	Ce code identifie l'heuristique à utiliser pour générer les parcours, il prend les valeurs suivantes:

code d'heuristique	heuristique
1	Clarke & Wright
2	Gillet & Miller
3	Newton & Thomas
4	Insertion
5	Yellow

Remarques	Description

(4) Le code de formule est en relation avec le code d'heuristique dans le choix de la formule d'évaluation. Le tableau suivant décrit la correspondance.

code d'heuristique	code de formule	formule
1	0	(4.5)'
1	2	(4.11)'
4	0	(6.2)
4	1	(6.3)
5	0	(4.5)'
5	2	(4.11)'

(5) Le code d'optimisation commande l'utilisation du OPT-2 ou du OPT-3 pour améliorer les parcours individuellement. A chaque fois que le module OPT-2 est appelé, celui-ci vérifie le code d'optimisation pour appeler un OPT-3 ou exécuter un OPT-2. Les codes se définissent comme suit:

code d'optimisation	type d'optimisation
2	OPT-2
3	OPT-3

Le OPT-2 est assumé par défaut.

Remarques **Description**

(6) Le code d'échange inter-parcours spécifie la demande d'une procédure d'échange inter-parcours de type OPT-2 ou l'absence complète de la procédure. La procédure appelée est dépendante du code d'origine des parcours. Si les parcours débutent et terminent à l'école on utilisera l'algorithme de la section 7.2.1; par contre si les parcours n'ont pas d'origine, on utilisera l'algorithme de la section 7.3.3. Le tableau suivant résume ceci. Si le code vaut 2 on applique la procédure sur tous les heuristiques traités. Si le code vaut 1 on applique la procédure seulement sur les heuristiques retenues.

code d'échange inter-parcours	code d'origine des parcours	algorithme utilisé
0		aucun
1 ou 2	1	7.2.1
1 ou 2	0	7.3.3

(7) Le code d'origine des parcours indique au programme que les parcours débutent à l'école ou que les parcours n'ont pas d'origine déterminée. Lorsque les parcours débutent et terminent à l'école, le réseau des arrêts est alors non-dirigé. Par contre, lorsque les parcours n'ont pas d'origine, le réseau devient dirigé par rapport à l'école, c'est-à-dire que la distance école-arrêt est considérée nulle. Le tableau suivant décrit ce code

code d'origine des parcours	origine des parcours	effet sur le réseau
0	indéterminée	dirigé par rapport à l'école
1	école	non-dirigé

Remarques | Description

(8) La valeur du paramètre λ doit être spécifiée pour les heuristiques de Clarke & Wright, Insertion formule (6.3) et Yellow. L'usager fournit la valeur initiale et finale de même que l'incrément. Ces trois valeurs sont en nombre réels, lues en format F5.3. Par exemple, si nous lisons λ initial 1.8, λ final 2.1 et incrément 0.1, le problème sera solutionné pour les valeurs de λ à 1.8, 1.9, 2.0, et 2.1. Le tableau suivant illustre la relation heuristique-paramètre utilisée dans nos productions.

code d'heuristique	code de formule	λ initial	λ final	incrément de λ
1	0	1.5	2.5	0.1
1	2	1.5	2.5	0.1
2	0	-	-	-
3	0	-	-	-
4	0	-	-	-
4	1	0.8	1.1	0.1
5	0	1.5	2.5	0.1
5	2	1.5	2.5	0.1

((-): aucun paramètre λ requis)

(9) Pour indiquer la fin de la liste des heuristiques à utiliser pour générer les parcours d'un problème, l'utilisateur doit fournir une carte dont le code d'heuristique est zéro. Lorsque cette carte est lue, le programme évalue la meilleure solution et réexécute les heuristiques dont les paramètres ont été retenus. Cette carte sert aussi de séparateur entre les paramètres de deux problèmes.

9.6.4 Description des sorties

Pour chacun des problèmes traités, l'usager reçoit une liste divisée en trois sections distinctes. La première section est unique et les sections deux et trois sont répétés pour chaque nombre de parcours différents obtenus dans la génération.

La première section contient la liste des heuristiques utilisés dans la génération des parcours. Pour chaque heuristique traité nous avons son code d'identification (cf. tableau 9.7) et son code de formule, son paramètre λ (PAR =), la distance totale des parcours construits (DIST TOT=), le nombre de parcours dans la solution (NB PARC =), l'estimé de la valeur de la solution (ESTIME = DIST TOT x NB PARC) et enfin le temps de résolution (TEMPS CPU =) excluant les entrées-sorties.

Tableau 9.7

code d'heuristique	heuristique
C&W	Clarke & Wright
G&M	Gillet & Miller
N&T	Newton & Thomas
INSR	Insertion
YLW	Yellow

La seconde section contient pour chaque solution retenue la description des contraintes du problème, l'identification de l'heuristique et des paramètres retenus, le nombre et la distance totale des parcours ainsi que les codes de bâtisses associés au problème. Pour chacun des parcours

numérotés à partir du nombre spécifié par l'usager, nous donnons le type et la capacité de l'autobus assigné, le nombre d'étudiants transportés, le nombre d'arrêts et la distance du parcours. Les noeuds des arrêts et de destinations sont ensuite listés en séquence de visite. Pour chaque arrêt, on donne le numéro du sommet, le noeud et le nombre d'étudiants embarqués.

La troisième section contient le tracé des parcours dans une matrice que l'on imprime. Pour chaque arrêt du réseau, nous donnons (dans la même séquence qu'ils apparaissent à la section deux) le numéro de sommet, le noeud, son abscisse et son ordonnée dans le tableau et le signe rattaché au parcours. Lorsque plus d'un signe est présent au même point, un message est émis et le signe est remplacé par (%). L'école est identifiée par (*).

*** SYSTEME DE TRANSPORT SCOLAIRE ***
*** GENERATION DES PARCOURS ***

COMMISSION SCOLAIRE ST-FRANCOIS (ARRETS PAR ORDINATEUR)

LISTE DES HEURISTIQUES DEMANDES ET RESULTATS

```
NOUS AVONS  1 TYPES D'AUTOBUS DISPONIBLES
CHACUNE NE PEUT FAIRE PLUS DE 20 ARRETS ET PARCOURIR PLUS DE  20000 METRES
LE PARAMETRE DE GENERATION DE PARCOURS UTILISE EST: 1.000

TYPE  1,100 AUTOBUS DISPONIBLES, DE CAPACITE  60 NOUS EN AVONS UTILISE  11

ALGORITHME:INSERTION
INSERTION TYPE: 1
OPTIMISATION DES PARCOURS PAR:OPT2
OPTIMISATION AVEC:OPT2
ECHANGE INTER-PARCOURS AVEC:OPT2 SPECIAL
CONSTRUCTION DES ROUTES SANS BOUCLES

CONSTRUCTION: 11 PARCOURS,  LA DISTANCE TOTALE PARCOURUE EST DE   56650 METRES
 POUR BATISSE(S)    71  44  60

PARCOURS # 301 AUTOBUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 59 FAIT 12 ARRETS ET PARCOURS  7840 METRES
LE PARCOURS EST COMME SUIT
                              [   3]        NOEUD #  16777988       (   4 ETUDIANTS)
                              [   2]        NOEUD #  16759971       (   4 ETUDIANTS)
                              [  93]        NOEUD #  197521112      (   3 ETUDIANTS)
                              [  95]        NOEUD #  197926140      (   5 ETUDIANTS)
                              [  94]        NOEUD #  197586341      (   1 ETUDIANTS)
                              [  97]        NOEUD #  197473467      (   8 ETUDIANTS)
                              [  90]        NOEUD #  187443044      (   2 ETUDIANTS)
                              [  89]        NOEUD #  187262037      (   6 ETUDIANTS)
                              [  88]        NOEUD #  187165140      (   7 ETUDIANTS)
                              [  87]        NOEUD #  186954411      (   5 ETUDIANTS)
                              [  86]        NOEUD #  186684559      (   7 ETUDIANTS)
                              [  85]        NOEUD #  186442632      (   7 ETUDIANTS)
                                            NOEUD #  174985778
                                            NOEUD #  163913050
                                            NOEUD #  163742207

PARCOURS # 302 AUTOBUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 60 FAIT 18 ARRETS ET PARCOURS 11420 METRES
LE PARCOURS EST COMME SUIT
                              [  13]        NOEUD #  151922318      (   1 ETUDIANTS)
                              [  12]        NOEUD #  151852372      (   4 ETUDIANTS)
                              [  14]        NOEUD #  151955506      (   1 ETUDIANTS)
                              [  15]        NOEUD #  152079065      (   8 ETUDIANTS)
                              [  16]        NOEUD #  152145628      (   2 ETUDIANTS)
                              [  17]        NOEUD #  152216798      (   3 ETUDIANTS)
                              [   4]        NOEUD #  141578293      (   3 ETUDIANTS)
                              [   5]        NOEUD #  142061548      (   7 ETUDIANTS)
                              [   6]        NOEUD #  142613105      (   1 ETUDIANTS)
                              [   7]        NOEUD #  142948129      (   3 ETUDIANTS)
                              [   9]        NOEUD #  143392362      (   1 ETUDIANTS)
                              [  10]        NOEUD #  143514152      (   1 ETUDIANTS)
                              [  19]        NOEUD #  155345978      (   1 ETUDIANTS)
                              [   8]        NOEUD #  143259160      (   2 ETUDIANTS)
                              [  18]        NOEUD #  155199765      (   7 ETUDIANTS)
                              [  22]        NOEUD #  155213607      (   6 ETUDIANTS)
                              [  11]        NOEUD #  143416437      (   2 ETUDIANTS)
                              [  20]        NOEUD #  155567469      (   7 ETUDIANTS)
                                            NOEUD #  163742207
                                            NOEUD #  163913050
                                            NOEUD #  174985778

PARCOURS # 303 AUTOBUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 60 FAIT  9 ARRETS ET PARCOURS  5696 METRES
```

[29] NOEUD # 163076020 (10 ETUDIANTS)
[30] NOEUD # 163076020 (8 ETUDIANTS)
[42] NOEUD # 173818705 (3 ETUDIANTS)
[41] NOEUD # 172738373 (5 ETUDIANTS)
[43] NOEUD # 173055340 (7 ETUDIANTS)
[44] NOEUD # 173150475 (2 ETUDIANTS)
[46] NOEUD # 174068071 (9 ETUDIANTS)
[45] NOEUD # 174068071 (11 ETUDIANTS)
[51] NOEUD # 174064542 (8 ETUDIANTS)
NOEUD # 174985778
NOEUD # 163913050
NOEUD # 163702207

PARCOURS # 304 AUTORUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 60 FAIT 10 ARRETS ET PARCOURS 6183 METRES
LE PARCOURS EST COMME SUIT
[33] NOEUD # 164192418 (1 ETUDIANTS)
[28] NOEUD # 154636074 (12 ETUDIANTS)
[27] NOEUD # 154440224 (8 ETUDIANTS)
[26] NOEUD # 154132545 (7 ETUDIANTS)
[23] NOEUD # 153851892 (7 ETUDIANTS)
[25] NOEUD # 153918458 (7 ETUDIANTS)
[24] NOEUD # 153918458 (9 ETUDIANTS)
[21] NOEUD # 153652164 (8 ETUDIANTS)
[32] NOEUD # 163864837 (8 ETUDIANTS)
[31] NOEUD # 163200578 (1 ETUDIANTS)
NOEUD # 163702207
NOEUD # 163913050
NOEUD # 174985778

PARCOURS # 305 AUTORUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 60 FAIT 7 ARRETS ET PARCOURS 3763 METRES
LE PARCOURS EST COMME SUIT
[35] NOEUD # 165114834 (11 ETUDIANTS)
[36] NOEUD # 165114834 (10 ETUDIANTS)
[40] NOEUD # 165721372 (8 ETUDIANTS)
[39] NOEUD # 165609220 (3 ETUDIANTS)
[38] NOEUD # 165366401 (12 ETUDIANTS)
[37] NOEUD # 165254250 (5 ETUDIANTS)
[34] NOEUD # 165005003 (11 ETUDIANTS)
NOEUD # 174985778
NOEUD # 163913050
NOEUD # 163702207

PARCOURS # 306 AUTORUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 54 FAIT 5 ARRETS ET PARCOURS 2864 METRES
LE PARCOURS EST COMME SUIT
[47] NOEUD # 174216286 (11 ETUDIANTS)
[49] NOEUD # 174315364 (11 ETUDIANTS)
[48] NOEUD # 174315364 (12 ETUDIANTS)
[50] NOEUD # 174064542 (8 ETUDIANTS)
[55] NOEUD # 174064709 (11 ETUDIANTS)
NOEUD # 174985778
NOEUD # 163913050
NOEUD # 163702207

PARCOURS # 307 AUTORUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 46 FAIT 4 ARRETS ET PARCOURS 3060 METRES
LE PARCOURS EST COMME SUIT
[58] NOEUD # 174555981 (8 ETUDIANTS)
[59] NOEUD # 174703823 (12 ETUDIANTS)
[60] NOEUD # 174816405 (13 ETUDIANTS)
[62] NOEUD # 175048458 (8 ETUDIANTS)
NOEUD # 174985778
NOEUD # 163913050
NOEUD # 163702207

PARCOURS # 308 AUTOBUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 55 FAIT 6 ARRETS ET PARCOURS 2654 METRES
LE PARCOURS EST COMME SUIT

(67)	NOEUD # 1756229B0	(4 ETUDIANTS)
(66)	NOEUD # 175510830	(11 ETUDIANTS)
(65)	NOEUD # 175360674	(9 ETUDIANTS)
(64)	NOEUD # 175360674	(10 ETUDIANTS)
(63)	NOEUD # 175244708	(11 ETUDIANTS)
(61)	NOEUD # 175044858	(10 ETUDIANTS)
	NOEUD # 174985778	
	NOEUD # 163913050	
	NOEUD # 163702207	

PARCOURS # 309 AUTOBUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 60 FAIT 6 ARRETS ET PARCOURS 3259 METRES
LE PARCOURS EST COMME SUIT

(76)	NOEUD # 1848549B6	(12 ETUDIANTS)
(53)	NOEUD # 174547041	(10 ETUDIANTS)
(52)	NOEUD # 174547081	(12 ETUDIANTS)
(58)	NOEUD # 174683267	(7 ETUDIANTS)
(57)	NOEUD # 174683267	(9 ETUDIANTS)
(56)	NOEUD # 174604709	(10 ETUDIANTS)
	NOEUD # 174985778	
	NOEUD # 163913050	
	NOEUD # 163702207	

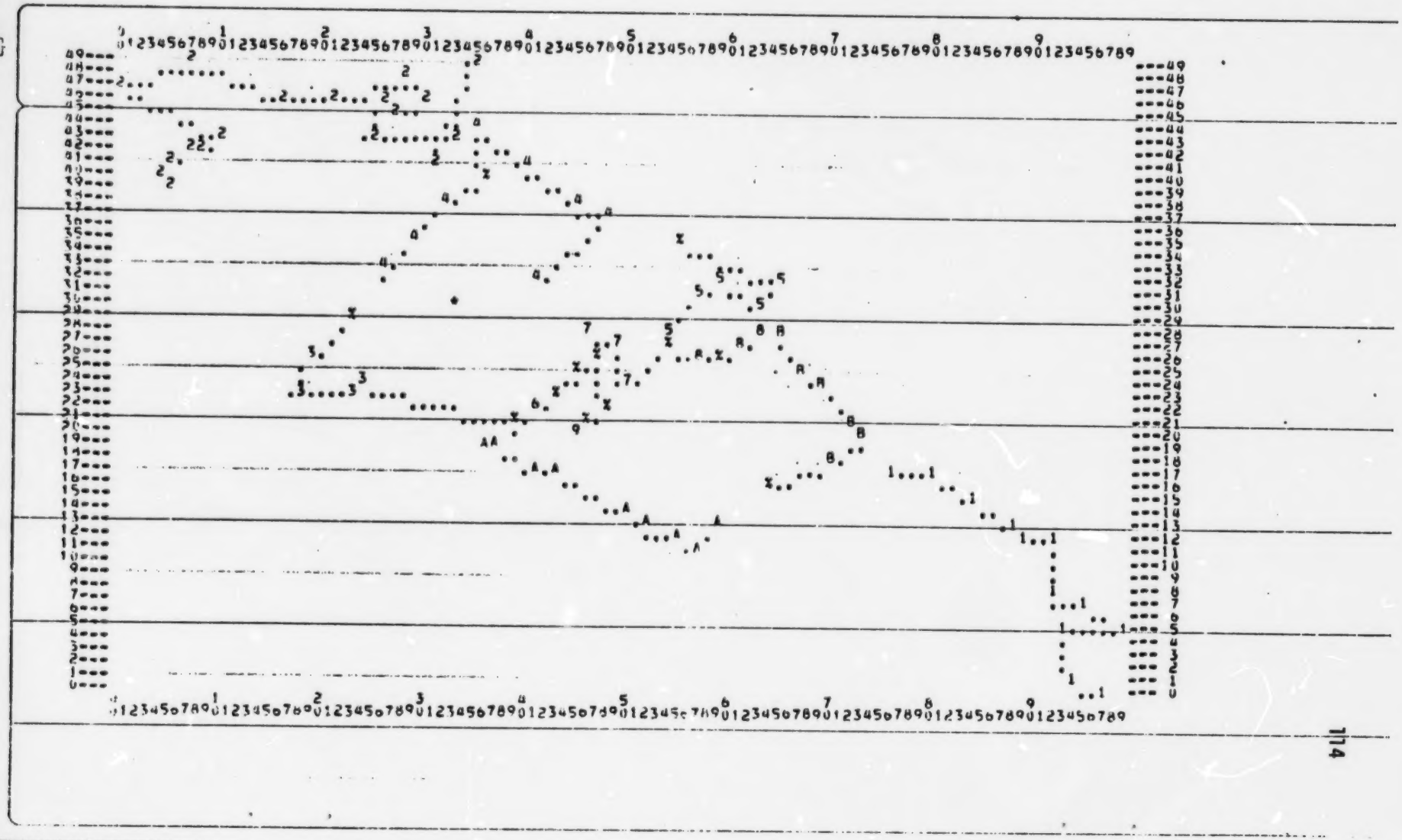
PARCOURS # 310 AUTOBUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 59 FAIT 9 ARRETS ET PARCOURS 5185 METRES
LE PARCOURS EST COMME SUIT

(80)	NOEUD # 1853BA102	(8 ETUDIANTS)
(91)	NOEUD # 19523211	(3 ETUDIANTS)
(79)	NOEUD # 185064022	(8 ETUDIANTS)
(78)	NOEUD # 184B60170	(8 ETUDIANTS)
(77)	NOEUD # 184759268	(8 ETUDIANTS)
(75)	NOEUD # 184371560	(10 ETUDIANTS)
(74)	NOEUD # 184230662	(3 ETUDIANTS)
(73)	NOEUD # 183902842	(8 ETUDIANTS)
(72)	NOEUD # 1A3924996	(15 ETUDIANTS)
	NOEUD # 174985778	
	NOEUD # 163913050	
	NOEUD # 163702207	

PARCOURS # 311 AUTOBUS DE TYPE # 1 DE CAPACITE 60 AVEC CHARGE 60 FAIT 8 ARRETS ET PARCOURS 4730 METRES
LE PARCOURS EST COMME SUIT

(81)	NOEUD # 1B5B5B494	(9 ETUDIANTS)
(82)	NOEUD # 1B5B5B494	(7 ETUDIANTS)
(83)	NOEUD # 1B0037095	(10 ETUDIANTS)
(84)	NOEUD # 1B0230957	(11 ETUDIANTS)
(71)	NOEUD # 176212124	(8 ETUDIANTS)
(70)	NOEUD # 175913412	(8 ETUDIANTS)
(69)	NOEUD # 175872614	(5 ETUDIANTS)
(68)	NOEUD # 175740890	(2 ETUDIANTS)
	NOEUD # 174985778	
	NOEUD # 163913050	
	NOEUD # 163702207	

```
    56   1748604502    45    25   (6)
CONFLIT AVEC PARCOURS SIGNE (3)
    55   174640709     47    26   (6)
    54   174555961     46    28   (7)
    59   174743423     49    27   (7)
    60   174816445     50    24   (7)
    62   175048856     58    27   (7)
    67   175422904     63    28   (8)
    64   175510843     61    27   (8)
    65   175300074     59    26   (8)
    60   175360624     52    26   (8)
CONFLIT AVEC PARCOURS SIGNE (8)
    63   175240708     57    26   (8)
    61   175084445     50    27   (8)
CONFLIT AVEC PARCOURS SIGNE (7)
    76   184654950     45    20   (9)
    53   174547081     46    21   (9)
    52   174547081     45    21   (9)
CONFLIT AVEC PARCOURS SIGNE (9)
    58   174643267     48    22   (9)
    57   174643267     48    22   (9)
CONFLIT AVEC PARCOURS SIGNE (9)
    56   174640709     47    26   (9)
CONFLIT AVEC PARCOURS SIGNE (6)
    81   145308102     59    13   (A)
    91   195252917     57    11   (A)
    72   185044432     55    12   (A)
    78   184848170     52    13   (A)
    77   184759248     50    14   (A)
    75   184371546     43    17   (A)
    74   184234062     41    17   (A)
    73   183902802     37    19   (A)
    72   183924895     36    19   (A)
    81   185034402     64    16   (B)
    82   185658494     64    16   (B)
CONFLIT AVEC PARCOURS SIGNE (8)
    83   186057095     70    13   (B)
    81   186230957     73    20   (B)
    71   176212126     72    21   (B)
    79   176013412     69    24   (B)
    68   175472014     67    25   (B)
    69   175746690     65    25   (B)
```



10. Tableaux et analyse des résultats

Dans cette section nous présentons les résultats obtenus avec les heuristiques sur trois problèmes réels de transport scolaire. De chacun des problèmes nous avons obtenu divers réseaux d'arrêts en variant la contrainte de marche des étudiants dans le problème d'assignation des étudiants aux arrêts, traité par G. Dulac [D2]. Nos résultats portent donc sur quinze réseaux d'arrêt différents. Nous analysons par la suite le facteur coût performance pour enfin suggérer une méthode d'approche au problème dans une implantation à grande échelle.

10.1 Les problèmes, contraintes et heuristiques utilisés

Dans notre implantation, nous avons généré les parcours de trois groupes d'écoles de la Commission Scolaire Régionale St-François couvrant le milieu urbain de la ville de Drummondville. Pour chaque groupe d'école nous avons traité plusieurs assignations d'étudiants aux arrêts. En référant au travail de G. Dulac, premièrement, nous avons assigné les étudiants aux arrêts choisis par le régisseur, puis nous avons choisi les arrêts en fixant la contrainte de marche des étudiants à des distances de 350, 300 et 225 mètres. Nous avons aussi traité l'assignation manuelle des étudiants aux arrêts par le régisseur. Le tableau 10.1 décrit les problèmes traités.

Tableau 10.1 Description des problèmes traités

BATISSES / Cas traités	PROBLEMES 70	71,44,60	72,41
Choix manuel d'arrêts et assignation manuelle	1-1	2-1	3-1
Choix manuel d'arrêts et assignation par ordinateur	1-2	2-2	3-2
Choix et assignation par ordinateur contrainte de marche = 350 m.	1-3	2-3	3-3
Choix et assignation par ordinateur contrainte de marche = 300 m.	1-4	2-4	3-4
Choix et assignation par ordinateur contrainte de marche = 225 m.	1-5	2-5	3-5

Les contraintes utilisées pour la génération des parcours sont uniformes pour chaque problèmes traités et nous ont été proposés par le régisseur de la commission scolaire. Nous avons un type d'autobus avec une capacité de 60 étudiants (nombre d'autobus non contraint), chaque parcours ne peut faire plus de 30 arrêts avec une distance de trajet inférieure à 20 kilomètres. Les arrêts doivent contenir au plus 15 étudiants, ceux excédant ce nombre doivent être découpé en arrêts artificiels. On considère un arrêt artificiel comme un arrêt indépendant de son arrêt original. On permettra à chaque arrêt de rejoindre les 10 arrêts les plus rapprochés. Les problèmes sont solutionnés avec tous les heuristiques selon le tableau 10.2. Il est à noter que les parcours sont générés sans boucle de départ à l'école et sans procédure d'échange inter-parcours.

Tableau 10.2 Description des heuristiques et paramètres λ utilisés

heuristique	formule	nombre de passes	valeurs de λ (si requis)
Clarke & Wright (C&W)	(4.5)'	11	λ = 1.5, 1.6, ..., 2.4, 2.5
Clarke & Wright (C&W)	(4.11)'	11	λ = 1.5, 1.6, ... 2.4, 2.5
Gillet & Miller (G&M)		1	
Newton & Thomas (N&T)		1	
Insertion (INSR)	(6.2)	1	
Insertion (INSR)	(6.3)	3	λ = 0.9, 1.0, 1.1
Yellow (YLW)	(4.5)'	11	λ = 1.5, 1.6, ... 2.4, 2.5
Yellow (YLW)	(4.11)'	11	λ = 1.5, 1.6, ... 2.4, 2.5

10.2 Tableaux des solutions obtenues

Nous présentons dans le tableau 10.3 les solutions obtenues et les temps de résolution de chacun des problèmes pour les heuristiques utilisés. A l'intersection heuristique-problème nous donnons la solution à distance minimum pour chaque nombre de parcours obtenus. Notons que les temps de résolution[1] sont donnés pour onze exécutions des heuristiques de Clarke & Wright et Yellow et pour trois exécutions de l'heuristique d'insertion formule (6.3).

Nous avons volontairement omis la discussion du paramètre λ dans les formules (4.5)', (4.11)' et (6.3). Il fut démontré par Golden [G3,G4] qu'un examen de plusieurs λ doit être fait pour chaque problème et qu'une valeur

1 Les problèmes ont été résolus sur l'ordinateur CYBER 173 et les temps excluent les entrées-sorties.

de λ générant les parcours à faible distance, lorsqu'appliquée à un autre problème peut donner une très mauvaise solution. Un balayage de plusieurs valeurs de λ doit donc être fait pour chaque problème.

L'examen de ces tableaux démontre une certaine consistance dans les solutions n-3, n-4 et n-5 pour chaque groupe d'école n = 1, 2, 3. De façon générale plus la contrainte de marche est petite dans le problème de G. Dulac [D2], plus le nombre d'arrêts est grand et plus la distance de parcours est élevée pour le même heuristique.

Bien qu'il ne nous soit pas possible de généraliser à partir d'un échantillon de seulement quinze problèmes obtenus à partir de la réorganisation des arrêts de trois groupes d'écoles, nous pouvons proposer une approche au problème dans une implantation à grande échelle dans le but de pondérer les facteurs coûts-performances. Le tableau 10-4 résume les solutions choisies suivant le critère défini à partir du produit du nombre de parcours et de la distance totale. On y remarque que Clarke & Wright a donné les "meilleures" solutions 8 fois sur 15, que l'insertion vient en deuxième avec 6 sur 15 et enfin Newton & Thomas avec 1 sur 15.

Comme nous l'avions déduit dans la section 5, l'approche de Newton & Thomas domine celle de Gillet & Miller puisqu'elle tient compte de la structure du réseau. Nous remarquons aussi que l'insertion domine de façon générale la méthode de Yellow tout comme nous l'avions escompté à la section 6.2. Nous retenons donc les heuristiques de Clarke & Wright, d'insertion et de Newton & Thomas.

Tableau 10.3 Sommaire des solutions heuristiques

Heuristique / Problème		C&W (4.5)'	C&W (4.11)'	G&M	N&T	INSR (6.2)	INSR (6.3)	YLW (4.5)'	YLW (4.11)'
1-1	10p.				60886		61629	60922	
	11p.	54907*	55202°					62247	63408
83A	12p/13p.			80175(13)		69556(12)	65837(12)		66175(12)
582E	t	6.4	6.4	3.5	5.9	6.0	17.6	25.1	19.7
1-2	10p.	52828°	52262*	68283	62736		54984	60522	56849
	11p.	54023	53964			61155		58536	57460
80A	12p.								62179
545E	t	6.2	6.3	3.0	3.5	5.6	17.0	20.4	19.1
1-3	10p.	51996°	50336*	63518	56989	56398	56329	60124	57492
	11p.							60883	
68A	12p.								
545E	t	5.4	5.8	2.0	3.0	4.9	13.9	17.7	20.0
1-4	10p.	52551*	53135°	69414	61211	56890	57852	60088	56177
	11p.	55331	54017					65292	59844
77A	12p.							73688	66684
545E	t	6.3	6.7	3.1	4.1	5.9	17.8	17.8	20.2
1-5	10p.	56290	56127°				55831*	64269	65325
	11p.	57185	56956		67416	62558		65110	65454
96A	12p.			75421				69610	
545E	t	8.2	9.1	5.61	6.22	9.02	25.13	27.19	33.15

p. = parcours, t = temps C.P.U., A = arrêts, E = étudiants
* = meilleur heuristique, o = second meilleur heuristique

Heuristique / Problème		C&W (4.5)'	C&W (4.11)'	G&M	N&T	INSR (6.2)	INSR (6.3)	YLW (4.5)'	YLW (4.11)'
2-1	11p.						56010*	61896	61022
	12p.	54858	54292°		66269	61991		66417	64697
94A	13p./14p.			75112(14)				70043(13)	67942(13)
628E	t	8.3	8.5	4.4	4.4	9.3	24.7	30.3	25.6
2-2	11p.			68321	62466	64747	55731*	60824	63084
	12p.	53689°	53689°					61990	63458
82A	13p.							71114	66604
595E	t	6.3	6.3	3.6	3.5	6.3	18.5	22.7	21.0
2-3	11p.				62774	60204°	57567*	68412	62825
	12p.		55245	70591				66234	66957
77A	13p.	57005	57005					68516	
596E	t	5.6	6.0	2.5	3.0	6.0	17.2	24.4	26.4
2-4	11p.				62568		57419*	61072	62852
	12p.	54884°	56591	70401		64575		64391	68166
85A	13p.							68194	66516
595E	t	6.2	7.4	4.6	3.8	7.4	19.5	25.7	27.0
2-5	10p./11p.	56203(11)°			60982(10)*		59516(11)	68071(11)	66039(11)
	12p.	59185	60155	73925		69450	64748	65235	70107
99A	13p.							68975	77300
585E	t	8.3	8.4	6.8	6.1	8.8	26.5	30.8	33.7

Heuristique / Problème		C&W (4.5)'	C&W (4.11)'	G&M	N&T	INSR (6.2)	INSR (6.3)	YLW (4.5)'	YLW (4.11)'
3-1	8p.	33662^{o*}	33662^{o*}	45254	43945		39039	42293	40155
	9p.					39484	40474	41143	43104
60A									
418E	t	4.3	4.4	3.6	2.7	4.2	12.0	12.1	14.4
3-2	8p.	33070^{o*}	33070^{o*}		37508	37032	36735	43466	40330
	9p.	33093	33093	39455			37370	42561	38753
57A									
390E	t	4.2	4.1	7.3	2.7	4.0	11.6	12.5	14.8
3-3	8p.	39369*	39615^{o}		41377		40077	42380	44358
	9p.		38400	45921		47346		45472	42612
55A									
391E	t	6.1	5.4	3.8	5.7	3.7	8.9	14.8	13.3
3-4	8p.	39278*	39839^{o}	50944	46864		43664	44510	46674
	9p.		43060				40781	44651	47818
62A	10p.					48468			
391E	t	6.3	8.1	3.4	3.7	4.8	13.2	21.4	15.5
3-5	8p.	43350^{o}	43350^{o}	48980	46150		42253*	50107	47971
	9p.	42210	43027				43147	52708	48173
70A	10p.					48234		52574	52592
391E	t	8.8	10.4	16.6	11.0	7.3	19.6	27.1	29.6

Dans le but de déduire une approche au problème, voyons maintenant le tableau 10.5 qui relate les deux meilleurs heuristiques (non pas solutions) selon notre critère parcours-distance. On y remarque la nette dominance de l'heuristique de Clarke & Wright avec les formules (4.5)' et (4.11)'. En nous basant sur la complexité des heuristiques, nous proposons une approche de résolution par étapes tel que décrite dans le schéma 10.7.

Tableau 10.4

TABLEAU DES SOLUTIONS CHOISIES

PROBLEME	HEURISTIQUE	NB.PARCOURS	DISTANCE TOTALE
1-1	C&W(4.5)'	11	54907
1-2	C&W(4.11)'	10	52262
1-3	C&W(4.11)'	10	50336
1-4	C&W(4.5)'	10	52551
1-5	INSR(6.3)	10	55831
2-1	INSR(6.3)	11	56010
2-2	INSR(6.3)	11	55731
2-3	INSR(6.3)	11	57567
2-4	INSR(6.3)	11	57419
2-5	N&T	10	60982
3-1	C&W(4.5)'&(4.11)'	8	33662
3-2	C&W(4.5)'&(4.11)'	8	33070
3-3	C&W(4.5)	8	39369
3-4	C&W(4.5)	8	39278
3-5	INSR(6.3)	8	42253

Tableau 10.5

TABLEAU DES DEUX "MEILLEURS" HEURISTIQUES

PROBLEME	PREMIER	SECOND
1-1	C&W(4.5)'	C&W(4.11)'
1-2	C&W(4.11)'	C&W(4.5)'
1-3	C&W(4.11)'	C&W(4.5)'
1-4	C&W(4.5)'	C&W(4.11)'
1-5	INSR(6.3)	C&W(4.11)'
2-1	INSR(6.3)	C&W(4.11)'
2-2	INSR(6.3)	C&W(4.5)'&(4.11)'
2-3	INSR(6.3)	INSR(6.2)
2-4	INSR(6.3)	C&W(4.5)'
2-5	N&T	C&W(4.5)'
3-1	C&W(4.5)'&(4.11)'	C&W(4.5)'&(4.11)'
3-2	C&W(4.5)'&(4.11)'	C&W(4.5)'&(4.11)'
3-3	C&W(4.5)'	C&W(4.11)'
3-4	C&W(4.5)'	C&W(4.11)'
3-5	INSR(6.3)	C&W(4.5)'&(4.11)'

Hiérarchie des fréquences

C&W(4.11)
C&W(4.5)
INSR(6.3)
N&T
INSR(6.2)

Tableau 10.5

COMPLEXITE DES HEURISTIQUES

Heuristique	Complexité *
Clarke & Wirght	$\theta(n\log_2 n + \frac{n^2}{p})$
Gillet & Miller	$\theta(n^2 + \frac{n^3}{p})$
Newton & Thomas	$\theta(n\log_2 n + \frac{n^3}{p})$
Insertion	$\theta(n^2\log_2 n + \frac{n^2}{p})$
Yellow	$\theta(\frac{n^2}{p}\log_2 n + \frac{n^2}{p})$
Echanges inter-parcours de type OPT-2	$\theta(n^2)$

n = nombre d'arrêts

p = nombre de parcours

(le nombre d'arcs est proportionnel au nombre d'arrêts)

* La complexité est estimé en fonction du nombre maximum d'opération requis pour solutionner, ces estimés sont inspirés du texte de Golden [G3] et adaptés au problème à plusieurs véhicules.

Schéma 10.7 Approche au problème

Etapes	Description
[A]	Générer les parcours avec l'heuristique de Clarke & Wright à l'aide des formules (4.5)' et (4.11)' pour diverses valeurs de λ.
[B]	Analyser les solutions obtenues et déterminer si le nombre de parcours devrait être réduit. Si non, aller à l'étape [E] Si oui, aller à l'étape [C]
[C]	Générer les parcours avec les heuristiques d'Insertion à l'aide de la formules (6.3) pour diverses valeurs de λ, et avec l'heuristique de Newton & Thomas.
[D]	Analyser les solutions obtenues aux étapes [A] et [C] et choisir une solution appropriée.
[E]	Si la solution ne respecte pas tous les critères du régisseur, apporter des solutions manuelles au parcours sélectionnés.
[F]	Enregistrer les parcours et terminer.

10.3 La post-optimisation des parcours

En nous basant sur l'article de Golden [G3] on atteint facilement sur un problème à un seul véhicule 5% de la meilleure solution connue en choisissant la meilleure solution heuristique. On peut réduire à 2% en appliquant une post-optimisation aux solutions heuristiques. En nous basant sur cette remarque et en généralisant au problème à plusieurs véhicules, nous avons choisi les heurisitques retenus à la section 10.2 et pour chacune des solutions obtenues nous avons appliqué la procédure d'échange inter-parcours de la section 7.3.3. Les sections suivantes relatent les résultats obtenus et proposent une utilisation de la post-optimisation qui tient compte du facteur coût-performance.

10.3.1 Choix des heuristiques et paramètres λ

Nous avons retenu pour la post-optimisation les heuristiques de Clarke & Wright, de Newton & Thomas et d'insertion selon les paramètres λ et formules décrites dans le tableau 10.8. Nous avons volontairement réduit le nombre de λ analysés puisque les solutions des heuristiques pour nos problèmes sont stables à des écarts de 0.2 dans λ et puisque la post-optimisation est de coûteuse complexité $\theta(n^2)$

Tableau 10.8

HEURISTIQUES A POST-OPTIMISER ET CHOIX DU PARAMETRE λ

Heuristique	formule	nombre de passes	valeurs de λ (si requis)
Clarke & Wright (C&W)	(4.5)'	5	λ=1.5,1.7,1.9,2.1, 2.3
Clarke & Wright (C&W)	(4.11)'	5	λ=1.5,1.7,1.9,2.1, 2.3
Newton & Thomas (N&T)		1	
Insertion (INSR)	(6.3)	3	λ=0.9,1.0,1.1

10.3.2 Tableaux des solutions obtenues

Nous présentons dans le tableau 10.9 les meilleures solutions obtenues et les temps de résolution de chaque heuristique retenu sous un format similaire au tableau 10.8. Les temps de résolution sont donnés pour cinq exécutions de l'heuristique Clarke & Wright et pour trois exécutions de l'heuristique d'insertion.

On remarque en comparant le tableau 10.9 au tableau 10.3 que l'heuristique de Clarke & Wright est le moins affecté par la post-optimisation mais que les heuristiques de Newton & Thomas et d'insertion offrent les meilleures réductions de distance. Ceci tient du fait que l'heuristique de Clarke & Wright construit ses parcours en minimisant la distance

totale. La méthode de Newton & Thomas est dépendante de la qualité du tour initial et l'insertion minimise localement un parcours à la fois. Ces dernières ont tendance à laisser peu d'étudiants au dernier parcours permettant ainsi plusieurs échanges autrement non-réalisables.

Le tableau 10.10 présente les solutions retenues pour l'agencement des parcours, on y remarque peu de différences par rapport au tableau 10.4, nous avons présenté dans le tableau 10.11 une comparaison des solutions retenues. La colonne de gauche donne les solutions retenues sans post-optimisation, celle de droite donne les solutions retenues avec post-optimisation et celle du centre nous donne la valeur et le rang des solutions obtenues dans la colonne de gauche auxquelles nous avons appliqué la post-optimisation. Cette post-optimisation de la meilleure solution heuristique nous classe dans nos quinze problèmes soit au premier ou second rang par rapport à la meilleure solution connue. En nous référant à la colonne %OPT, pour les solutions de même nombre de parcours les solutions heuristiques sont au plus à 7.3% de la meilleure solution et les solutions heuristiques post-optimisées sont au plus à 1.7% de la meilleure solution, ce qui vérifie sur nos problèmes les résultats obtenus par Golden [G3] sur le problème à un seul véhicule. Il est donc facile d'obtenir près de 5% de la meilleure solution et près de 2% en appliquant la post-optimisation aux solutions retenues pour les problèmes à n véhicule que nous avons traité.

Dans une implantation à grande échelle, la post-optimisation de chacune des solutions produites serait coûteuse. L'utilisation verra donc à pondérer les facteurs coûts-performances en n'appliquant la post-

optimisation qu'aux heuristiques faibles comme Newton & Thomas et Insertion ou/et qu'aux solutions retenues. Nous recommandons toutefois de ne pas omettre la post-optimisation de la meilleure solution heuristique.

Tableau 10.9

SOMMAIRE DES SOLUTIONS HEURISTIQUES POST-OPTIMISEES

Heuristique / Problème		C&W (4.5)'	C&W (4.11)'	N&T	INSR (6.3)
(1-1)'	10p.			59414*	60065
83A	11p.	54907°	55202		
582E					
	t	29.7	27.8	12.8	39.5
(1-2)'	10p.	51283*	52262°	60355	54324
80A	11p.	53192	53964		
545E					
	t	48.2	25.7	9.6	38.0
(1-3)'	10p.	51868°	50029*	54606	53217
68A	11p.		52095		
545E					
	t	24.4	34.1	9.5	33.1
(1-4)'	10p.	52551*	52837°	59573	55729
77A	11p.		54324		
545E					
	t	25.9	28.3	12.7	35.5
(1-5)'	10p.	56127°	56127°		55484*
96A	11p.	57558	56384	66086	
545E					
	t	35.8	35.0	15.0	52.6

p = parcours, t = temps C.P.U., A = arrêts, E = étudiants

* = meilleur heuristique, o = second meilleur heuristique

Tableau 10.9 (suite)

Heuristique / Problème		C&W (4.5)'	C&W (4.11)'	N&T	INSR (6.3)
(2-1)'	11p.				55809*
94A	12p.	54858	54292°	62510	
628E					
	t	34.0	34.0	14.4	45.0
(2-2)'	11p.	54363*		61392	54579°
82A	12p.	53326	53326		
595E					
	t	39.8	36.5	9.2	42.0
(2-3)'	11p.			61981	56770*
	12p.	55272	54969°		
77A	13p.	56944			
596E					
	t	38.8	38.9	10.7	36.7
(2-4)'	11p.			59640°	55155*
85A	12p.	54883	55892		
595E					
	t	39.7	36.1	12.9	45.2
(2-5)'	10p.			60982*	
	11p.	55848°			
99A	12p.	58941	59895		57874
585E					
	t	47.8	51.7	12.1	56.0

Tableau 10.9 (suite)

Heuristique / Problème		C&W (4.5)'	C&W (4.11)'	N&T	INSR (6.3)
(3-1)'	8p.	33662*°	33662*°	38693	37890
60A	9p.				38879
418E					
	t	14.5	14.5	8.6	22.8
(3-2)'	8p.	30816*°	30816*°	36561	33974
57A	9p.				35866
390E					
	t	23.7	23.7	5.3	23.2
(3-3)'	8p.	37423°	36942*	39994	38062
55A	9p.		38400		
391E					
	t	17.5	17.6	9.2	19.7
(3-4)'	7p.				41336*
62A	8p.	38853°	39695	46863	40701
391E	9p.				40635
	t	24.62	17.8	6.5	27.1
(3-5)'	8p.	41053°	43349	42859	40515*
70A	9p.	42209	42270		43001
391E					
	t	30.0	26.0	17.7	34.7

Tableau 10.10

TABLEAU DES SOLUTIONS POST-OPTIMISEES CHOISIES

Problème	Heuristique	NB. Parcours	Distance totale
1-1'	N&T	10	59414
1-2'	C&W (4.5)'	10	51283
1-3'	C&W (4.11)'	10	50029
1-4'	C&W (4.5)'	10	52551
1-5'	INSR (6.3)	10	55484
2-1'	INSR (6.3)	11	55809
2-2'	C&W (4.5)'	11	54363
2-3'	INSR (6.3)	11	56770
2-4'	INSR (6.3)	11	55155
2-5'	N&T	10	60982
3-1'	C&W (4.5)' & (4.11)'	8	33662
3-2'	C&W (4.5)' & (4.11)'	8	30816
3-3'	C&W (4.11)'	8	36942
3-4'	INSR (6.3)	7	41336
3-5'	INSR (6.3)	8	40515

Tableau 10.11

COMPARAISON AVEC/SANS POST-OPTIMISATION

	Heuristique seulement			Post-optimisation du meilleur heuristique				Heuristique Post-optimisation		
	N.P.	DIST.	% OPT	N.P.	DIST	RANG	% OPT	N.P.	DIST	% OPT
1-1	11	54907	-	11	54907	2	-	10	59414	0%
1-2	10	52262	2%	10	52262	2	2%	10	51283	0%
1-3	10	50336	.6%	10	50029	1	0%	10	50029	0%
1-4	10	52551	0%	10	52551	1	0%	10	52551	0%
1-5	10	55831	.6%	10	55484	1	0%	10	55484	0%
2-1	11	56010	.3%	11	55809	1	0%	11	55809	0%
2-2	11	55731	2.5%	11	54579	2	.4%	11	54363	0%
2-3	11	57967	1.4%	11	56770	1	0%	11	56770	0%
2-4	11	57419	4%	11	55155	1	0%	11	55155	0%
2-5	10	60982	0%	10	60982	1	0%	10	60982	0%
3-1	8	33662	0%	8	33662	1	0%	8	33662	0%
3-2	8	33070	7.3%	8	31355	2	1.7%	8	30816	0%
3-3	8	39369	6.6%	8	37423	2	1.3%	8	36942	0%
3-4	8	39278	-	8	38853	2	-	7	41336	0%
3-5	8	42253	4.2%	8	40515	1	0%	8	40515	0%

OPT : Ecart par rapport à la meilleure solution obtenue

RANG: Rang par rapport à la meilleure solution

11. Le problème général sur les arrêts et segments

On ne possède que très peu d'algorithmes pour traiter le problème qui consiste à visiter des arrêts et à parcourir des segments décrit par Orloff [O1] comme le problème général de génération des parcours. En effet, on contourne en général la difficulté en transformant le problème en un problème sur des arrêts pour lequels des heuristiques efficaces existent. Dans cette section, nous proposons deux heuristiques pour traiter le problème général. Ceux-ci sont fortement inspirés des heuristiques de Clarke & Wright et d'insertion décrits aux chapitres 4 et 6 respectivement.

11.1 Transformation du problème

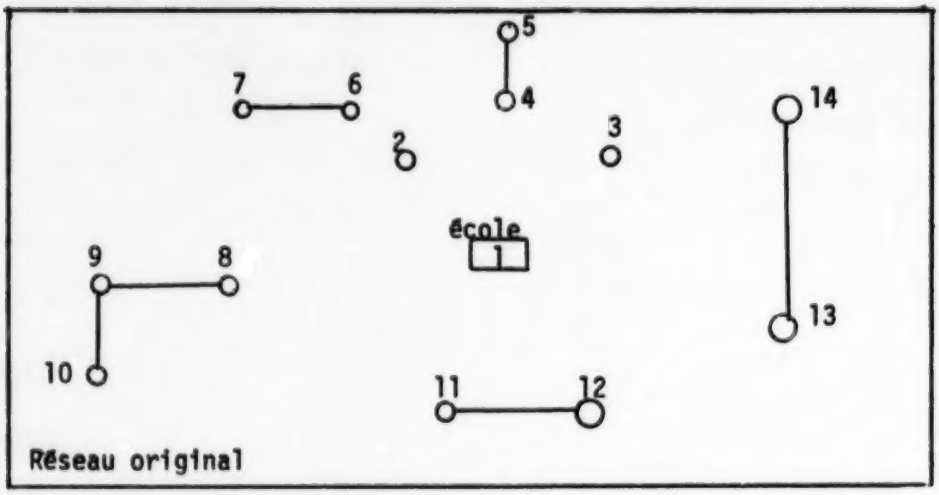
Selon Orloff [O1] le problème général de la génération de parcours sur les arrêts et segments peut se ramener à un problème plus simple sur les segments. Chaque arrêt i du réseau peut être transformé en un segment artificiel (i,i) avec distance nulle. La figure 11.1 illustre ce fait.

Le problème sur les segments est connu sous le nom de "postier chinois" ou "postier rural" selon qu'on doit ou non parcourir tous les segments du réseau. Si un seul véhicule est utilisé alors la méthode de solution est la suivante. Si chaque noeud est de degré pair, alors le tour minimum est de distance égale à la somme des distances de chaque segment. Ce tour n'est alors rien d'autre qu'un cycle d'Euler. Si les noeuds ne sont pas tous de degré pair, nous devons introduire des arcs artificiels (constitués de segments du réseau) pour relier les noeuds de degré impair. Une méthode optimale de solution par "matching" de noeuds est proposée par Edmonds [E1], un heuristique pour résoudre ce problème est aussi présenté

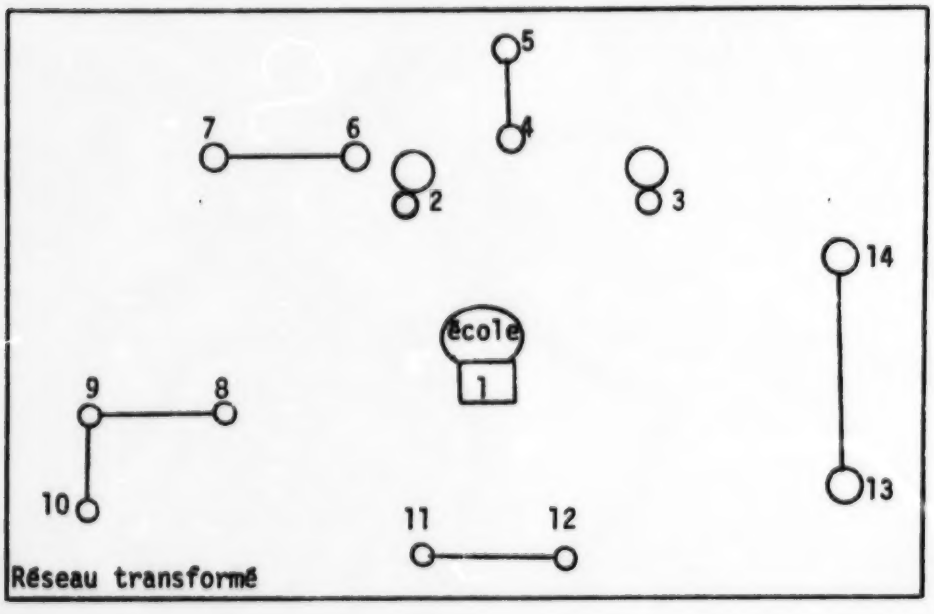
par Christofides [C3]. Il serait intéressant, dans l'approche "circuler en premier, grouper en second" de générer un grand tour parmi tous les segments à balayer, pour ensuite le découper en parcours réalisables.

Notre approche est celle de "grouper en premier, circuler en second" efficace en milieu urbain. Les heuristiques proposés sont décrits dans les sections ci-après.

figure 11.1 Transformation d'un problème sur arrêts et segments en un problème sur les segments



Arrets	Segments
1*	
2	
3	
	(4.5)
	(6.7)
	(8.9)
	(9.10)
	(11.12)
	(13.14)



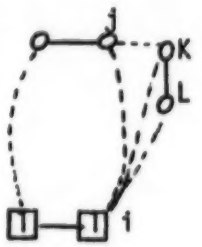
Segments
(1.1)
(2.2)
(3.3)
(4.5)
(6.7)
(8.9)
(9.10)
(11.12)
(13.14)

(*) l'école doit elle aussi être transformée en un segment

11.2 Heuristique d'Insertion - version segments

L'heuristique d'insertion sur les segments est très rapproché de l'heuristique décrit dans la section 6.3 à la différence près qu'il insère au parcours en construction non plus un seul arrêt mais bien un segment dans une certaine direction. Le parcours initial est constitué du segment école relié au segment possédant le noeud le plus éloigné de l'école. Le parcours est par la suite

figure 11.2



agrandit en insérant un segment (K,L) entre deux segments du parcours. Un segment (K,L) est considéré seulement si son introduction dans le parcours conserve la réalisibilité de ce dernier en termes de distance, de nombre de segments et de capacité. La désirabilité de l'introduction de chaque segment dans le parcours est évaluée par une formule S(i,K,L,j) où (K,L) est inséré entre les noeuds i et j du parcours. Une telle formule peut être (11.3)

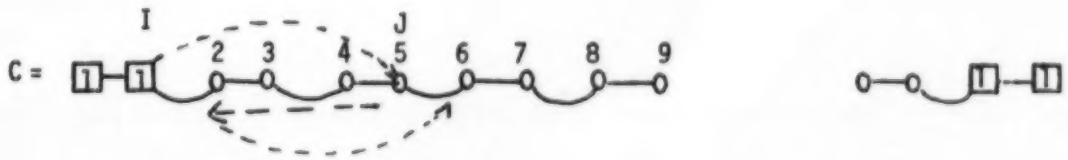
$$S(i,K,L,j) = d_{i,K} + d_{i,L} + d_{i,j} - \lambda [\, d_{j,K} + d_{i,L} \,] \qquad (11.3)$$

où λ est un paramètre permettant de varier les solutions.

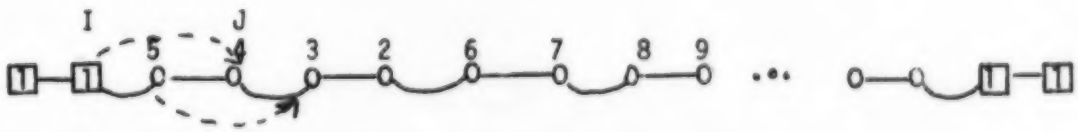
11.2.1 Post-optimisation des parcours

Lorsqu'un parcours est construit, nous pouvons tenter de diminuer sa distance en appliquant une procédure OPT-2 ou OPT-3 qui ne pourrait échanger que des arcs sans altérer les segments. Supposons le vecteur C contenant les noeuds du parcours, nous pouvons appliquer à ce vecteur une procédure OPT-2 en ne considérant que les noeuds I et J d'indice pair dans la séquence du vecteur C. Cette méthode est aussi valable pour le OPT-3 et nous assure de ne pas modifier les segments, mais seulement les arcs. La figure (11.4) illustre ce fait.

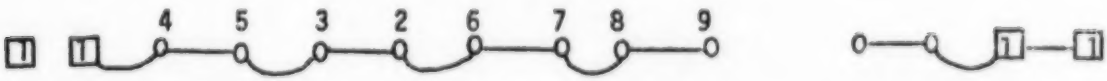
Figure 11.4



analyse des inversions sur I et J pairs seulement



Inversion de plusieurs segments



Inversion d'un segment

Nous pouvons généraliser cette remarque aux heuristiques d'échanges inter-parcours décrits dans les sections 7.2.1 et 7.3.3.

11.2.2 Etapes logiques de l'heuristique d'insertion-version segments

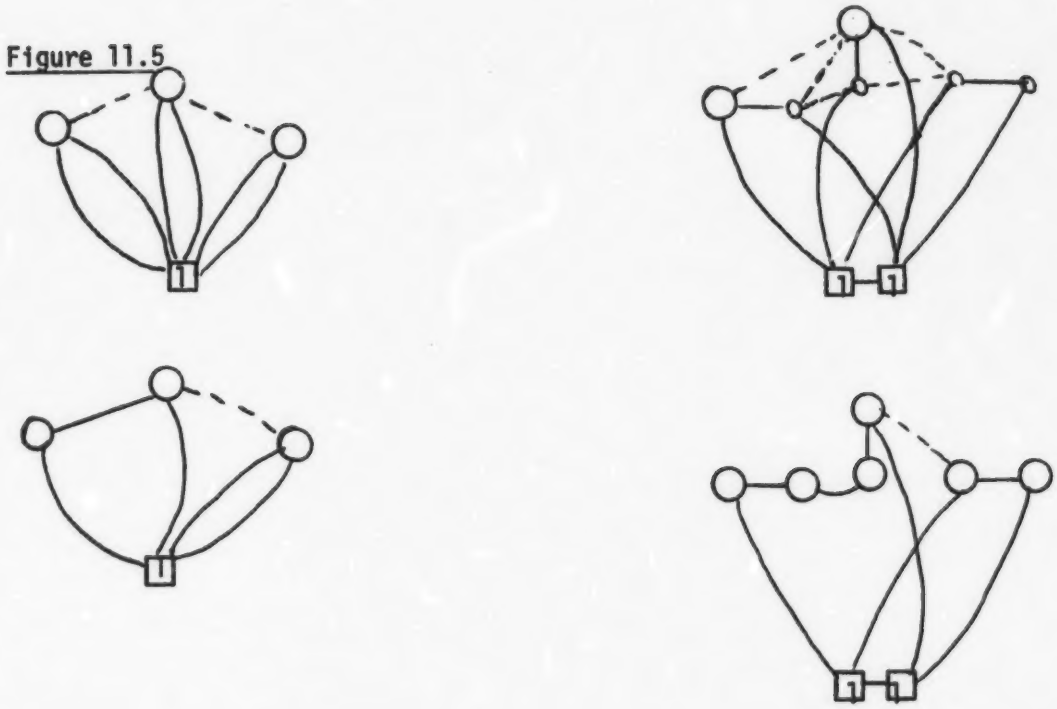
Etapes	Description
[A]	Trouver parmi la liste des segments hors-parcours celui dont un des deux noeuds est le plus éloigné de l'école. S'il n'existe plus de segment hors-parcours, aller à l'étape [H]
[B]	Former un cycle initial 1-(K,L)-1 où (K,L) est le segment sélectionné dans l'étape [A]. Retirer le segment (K,L) de la liste des segments hors-parcours.
[C]	Pour chacun des segments (K,L) hors-parcours trouver entre quel arc (i,j) (reliant deux segments) et dans quelle direction (K,L) ou (L,K) pourrait être inséré tel que son insertion dans le parcours respecte les contraintes de distance, de nombre de segments et de capacité. Choisir le quadriplet (i,K,L,j) tel que la formule 11.3 (avec λ fixée) soit minimum.
[D]	Parmi tous les (i,K,L,j) identifiés à l'étape [C] choisir (i*,K*,L*,j*) tel que S(i*,K*,L*,j*) de la formule (11.3) est minimum. S'il n'existe pas de tel (i*,K*,L*,j*) aller à l'étape [F]
[E]	Insérer le segment (K*,L*) entre les noeuds i* et j* du parcours. Retirer (K*,L*) de la liste des segments hors-parcours. Aller à l'étape [C].
[F]	Faire un OPT-2 (spécial pour les segments) sur le parcours généré. Si l'on ne réduit pas la distance, aller à l'étape [G]. Est-ce que l'on avait bloqué sur une contraite de distance?

Etapes	Description
	- si non aller à l'étape [G]
	- si oui aller à l'étape [C]
[G]	Enregistrer le parcours actuel.
	Aller à l'étape [A]
[H]	Faire une post-optimisation d'échange de segments entre les parcours de type OPT-2. (Adaptation des algorithmes 7.2.1 ou 7.3.3 pour les segments)
[I]	Terminer

11.3 Clarke & Wright - version segment

L'adaptation de l'heuristique de Clarke & Wright au problème sur les segments est la plus directe. Dans la méthode originale chaque arrêt est relié à l'école par un parcours et d'itérations en itérations les parcours sont combinés un à un pour en réduire le nombre. Dans le problème sur les segments, les segments sont reliés à l'école par un parcours et la même procédure de réduction de parcours peut être utilisée. La figure 11.5 illustre ceci.

Figure 11.5



ARRETS SEGMENTS

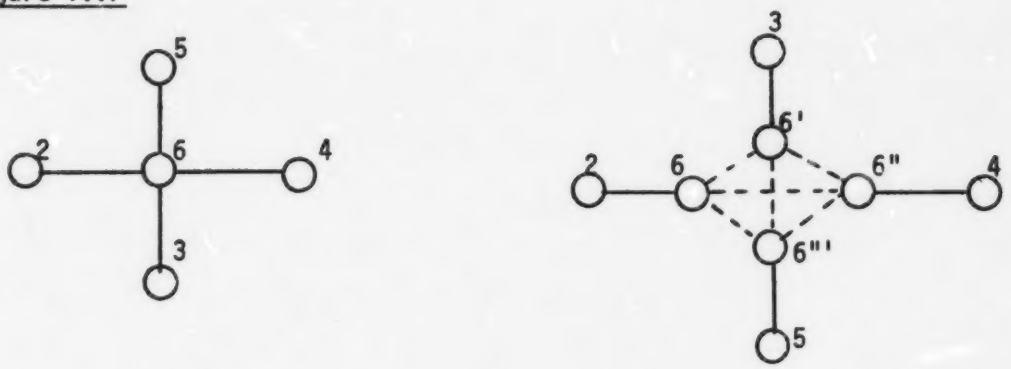
Exemple de réduction de parcours.

Dans la version segments les épargnes S_{ij} peuvent être calculés selon la formule originale 11.6, mais

$$S_{ij} = d_{1i} + d_{1j} - \lambda d_{ij} \qquad (11.6)$$

certains problèmes peuvent survenir à une intersection où plusieurs segments partagent le même noeud. Le noeud intersection doit être duplicaté autant de fois que de segments auquel il appartient.

Figure 11.7



On remarque dans la figure (11.7) qu'en employant la formule d'épargne (11.6) que

$$S6.6' = S6.6'' = S6.6''' = S6',6'' = S6',6''' = S6'',6'''$$

d'où une politique de sélection doit être mise sur pied. Il est possible de générer d'autres formules d'épargnes qui tiennent compte de la longueur des segments à coupler.

L'algorithme présenté à la section suivante assume implicitement les noeuds intersection duplicaté sans explicitement le faire. Comme dans l'heuristique (11.2.2) les segments sont considérés comme des entités individuelles auxquelles nous devons assigner un prédécesseur, un successeur et une direction de visite.

11.3 Etapes logiques de l'heuristique Clarke & Wright - version segments

Etapes	Description
[A]	Identifier les noeuds origine et destination de chaque segment i par (K_i, L_i)
[B]	Pour chaque paire de segments (K_i, L_i) et (K_j, L_j) reliés par au moins un arc dans le réseau des arcs et segments, évaluer et conserver

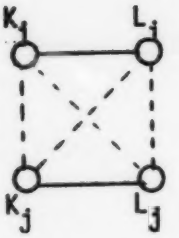
Etapes Description

$S(K_i,K_j)$ si (K_i,K_j) existe ou $K_i = K_j$

$S(K_i,L_j)$ si (K_i,L_j) existe ou $K_i = L_j$

$S(L_i,K_j)$ si (L_i,K_j) existe ou $L_i = K_j$

$S(L_i,L_j)$ si (L_i,L_j) existe ou $L_i = L_j$



[C] Ordonner les épargnes en ordre décroissant et additionner une constante M à chacune d'elle pour qu'elles soient toutes positives.

[D] Retirer de la liste la paire de segment i et j ayant la plus grande valeur d'épargne et dénotons cette épargne par S(i,j).

[E] Si $S(i,j) \leq 0$, aller à l'étape [H].

Si les segments i et j ne sont pas extrêmes, ou s'ils ne peuvent être réunis avec cette épargne, alors S(i,j) = 0 et aller à l'étape [D].

Si parcours (i) = parcours (j) alors

S(i,j) = 0

Aller à l'étape [D] .

Si nombre de segments (parcours (i)) + nombre de segments (parcours (j)) > maximum de segments

alors S(i,j) = 0

Aller à l'étape [D] .

Si charge (parcours (i)) + charge (parcours (j)) viole la contrainte de capacité, alors S(i,j) = 0

Aller à l'étape [D].

[F] Evaluer la distance du nouveau parcours où les parcours (i) et parcours (j) sont réunis.

Etapes	Description
	Si la distance excède la contrainte de distance, alors S(i,j) = 0
	Aller à l'étape [D]
[G]	Créer le nouveau parcours en reliant le parcours (i) et le parcours (j).
	Mettre à jour . la distance
	. le nombre de segments
	. la charge du nouveau parcours
	S(i,j) = 0
	Aller à l'étape [D]
[H]	Faire un OPT-2 (spécial pour les segments) sur chaque parcours générés
[I]	Faire une post-optimisation d'échange de segments entre les parcours de type OPT-2 (adaptation des algorithmes 7.2.1 ou 7.3.3 pour les segments)
[J]	Terminer.

12. Conclusion

En considérant l'adaptation des heuristiques de transport de marchandises pour construire des parcours uni-directionnels d'autobus, nous avons remarqué sur les problèmes traités que la performance des heuristiques est dépendante des réseaux sur lesquels on les appliquent. L'approche "grouper en premier, circuler en second" (Clarke & Wright, Insertion) a dominé de façon générale les résultats, toutefois certains problèmes ont mieux profités de l'approche "circuler en premier, grouper en second" (Newton & Thomas), cette dernière est toutefois dépendante de son cycle initial. La méthode de Clarke & Wright semble la plus robuste pour minimiser la distance totale mais souvent au détriment du nombre de parcours. Les méthodes de Newton & Thomas et d'insertion lorsque couplées à une post-optimisation inter-parcours deviennent très acceptables puisqu'elles ont déjà tendance à minimiser le nombre de parcours. Nous avons constaté que la meilleure solution heuristique nous amène, en ne considérant que le facteur de distance, à près de 5% de la meilleure solution connue, et que la post-optimisation de cette seule solution réduit l'écart à près de 2%. Ceci corrobore les résultats de Golden [G3] sur le problème à un seul véhicule. L'algorithme de post-optimisation par échanges inter-parcours de type OPT-2 (7.3.3) réduit de façon générale la distance totale et de façon indirecte le nombre de parcours, cette méthode est donc très valable et se doit d'être utilisée pour améliorer la solution heuristique retenue.

Dans l'extension de la recherche au problème sur les arrêts et segments, les algorithmes OPT-2 et OPT-3, de même que ceux de post-

optimisation pourront être utilisés moyennant de légères modifications. Les heuristiques proposés au chapitre 11 devraient être efficaces lorsque l'ensemble des segments à traverser forme un sous-ensemble restreint, non-connexe, de l'ensemble des segments du réseau rural.

Nous espérons enfin que notre contribution à l'automatisation du transport scolaire justifiera dans un avenir rapproché les efforts requis à sa réalisation par une implantation à l'échelle du Québec.

BIBLIOGRAPHIE

A1 R.D. Angel, W.L. Caudle, R. Noonan, and A. Whinston, "Computer-assisted School Bus Scheduling", Management Science, Series B, 18, (1972), 279-288.

B1 B. Bennet and D. Gazis, "School Bus Routing by Computer", Transportation Research, 6, (1972), 317-326.

B2 L. Bodin and L. Berman, "Routing and Scheduling of School Buses by Computer", Presented at Fall 1974 ORSA Meeting in San Juan.

C1 N. Christofides and S. Elion, "Algorithms for Large-scale Travelling Salesman Problems", Operational Research Quarterly, 23, (1972), 511-518.

C2 N. Christofides and S. Elion, "An Algorithm for the Vehicle-dispatching Problem", Operational Research Quarterly, 20, (1969), 309-318.

C3 N. Christofides, "The Optimum Traversal of a Graph", Omega, 1, (1973), 719-732.

C4 N. Christofides, "The Vehicle Routing Problem", RFAIRO (France), 10, (1976), 55-70.

C5 G.A. Croes, "A Method for Solving Travelling-Salesman Problems", Operations Research, 6, (1958), 791-812.

C6 G. Clarke and J.W. Wright, "Scheduling of Vehicules from a Central Depot to a Number of Delivery Points", Operations Research, 12, (1964), 568-581.

D1 J. Desrosiers, J.A. Ferland, and J.-M. Rousseau, "Horaires d'écoles et circuits d'autobus scolaires", Document de travail #88, Département d'informatique et de recherche opérationnelle, Université de Montréal, février 1976.

D2 G. Dulac, "Affectation des étudiants aux arrêts d'un système de transport des écoliers", Rapport de stage de maîtrise, Département d'informatique et de recherche opérationnelle, Université de Montréal, juin 1978.

E1 J. Edmonds and E.L. Johnson, "Matching, Euler tours and the Chinese Postman", Mathematical Programming, 5, (1973), 88-124.

F1 L. Ford and D. Fulkerson, "Flows in Networks", Princeton University Press, Princeton, (1962).

F2 B.L. Fox, "Data Structures and Computer Science Techniques in Operations Research", Publication #245, Département d'informatique et de recherche opérationnelle, Université de Montréal, décembre 1976.

G1 H. Gabbay, "An Overview of Vehicular Scheduling Problems", Technical report #103, M.I.T. (OSP 81138), September 1974.

G2 B.E. Gillet and L.R. Miller, "A Heuristic Algorithm for the Vehicle-Dispatch Problem", Operations Research, 22, (1974), 340-349.

G3 B. Golden, L. Bodin, T. Doyle, and W. Stewart,Jr., "Approximate Travelling Salesman Algorithms", Technical Report 77-005, College of Business and Management, University of Maryland, May 1977.

G4 B. Golden, T. Magnanti, and H. Nguyen, "Implementing Vehicle Routing Algorithms", Networks, 7, (1977), 113-148.

I1 IBM Corporation, IBM Vehicles Scheduling Program-Extended (VSP-X), Program Description Manual, report SHL19-2304-0, White Plains, New York (1968).

J1 J.G. Jacques, "Transport Scolaire, Analyse fonctionnelle, Module Localisation, Version 2, Phase 1", Commission Scolaire Régionale de Tilly, Sainte-Foy, Québec, 1976.

L1 N. Laplante et P.A. Blouin, "Guide du Transport Scolaire", Comité régional des transports, Ministère des transports, Québec, Avril 1973.

L2 S. Lin, "Computer Solutions of the Travelling-Salesman Problem", Bell System Technical Journal, 44, (1965), 2245-2269.

L3 S. Lin and B. Kernighan, "An Effective Heuristic Algorithm for the Travelling Salesman Problem", Operations Research, 21, (1973), 498-516.

M1 P. Martin et P. Vinet, "Transport scolaire, analyse préalable, Commission scolaire régionale de Tilly, septembre 1974.

N1 R.M. Newton and W.H. Thomas, "Design of School Bus Routes by Computer", Socio-Economic Planing Science, 3,(1969), 75-85.

N2 R.M. Newton and W.H. Thomas, "Bus Routing in a Multi-School System", Computer and Operations Research, 1, (1974), 213-222.

O1 C. Orloff, "A Fundamental Problem in Vehicle routing", Networks, 4, (1974), 35-64.

O2 C. Orloff, "Routing a fleet of m Vehicles to/from a Central Facility", Networks, 4, (1974), 147-162.

R1 R.A. Russel, "An Effective Heuristic for the M-tour Travelling Salesman Problem with some Side Conditions", Operations Research, 25, (1977), 517-524.

Y1 P.C. Yellow, "A Computational Modification to the Savings Method of Vehicle Scheduling", Operational Research Quarterly, 21, (1970), 28--283.